Push technology

Push technology, or **server push**, is a style of Internet-based communication where the request for a given transaction is initiated by the publisher or central server. It is contrasted with pull/get, where the request for the transmission of information is initiated by the receiver or client.

Push services are often based on information preferences expressed in advance. This is called a publish/subscribe model. A client "subscribes" to various information "channels" provided by a server; whenever new content is available on one of those channels, the server pushes that information out to the client.

Push is sometimes emulated with a polling technique, particularly under circumstances where a real push is not possible, such as sites with security policies that require rejection of incoming HTTP/S requests.

Contents

General use

Synchronous conferencing and instant messaging are typical examples of push services. Chat messages and sometimes files are pushed to the user as soon as they are received by the messaging service. Both decentralised peer-to-peer programs (such as WASTE) and centralised programs (such as IRC or XMPP) allow pushing files, which means the sender initiates the data transfer rather than the recipient.

Email may also be a push system: The SMTP protocol is a push protocol (see Push e-mail). However, the last step—from mail server to desktop computer—typically uses a pull protocol like POP3 or IMAP. Modern e-mail clients make this step seem instantaneous by repeatedly polling the mail server, frequently checking it for new mail. The IMAP protocol includes the IDLE command, which allows the server to tell the client when new messages arrive. The original BlackBerry was the first popular example of push-email in a wireless context.

Another example is the PointCast Network, which was widely adopted in the 1990s. It delivered news and stock market data as a screensaver. Both Netscape and Microsoft integrated push technology through the Channel Definition Format (CDF) into their software at the height of the browser wars, but it was never very popular. CDF faded away and was removed from the browsers of the time, replaced in the 2000s with RSS (a pull system.)

Other uses of push-enabled web applications include software updates distribution ("push updates"), market data distribution (stock tickers), online chat/messaging systems (webchat), auctions, online betting and gaming, sport results, monitoring consoles, and sensor network monitoring.

Examples

Webpush

The Webpush proposal of the Internet Engineering Task Force is a simple protocol using HTTP version 2 to deliver realtime events, such as incoming calls or messages, which can be delivered (or “pushed”) in a timely fashion. The protocol consolidates all real-time events into a single session which ensures more efficient use of network and radio resources. A single service consolidates all events, distributing those events to applications as they arrive. This requires just one session, avoiding duplicated overhead costs.[1]

Web Notifications are part of W3C standard and defines an API for end-user notifications. A notification allows alerting the user outside the context of a web page of an occurrence, such as the delivery of email. [2] As part of this standard Push API defined by W3C, is now being implemented by Chrome, Firefox, Edge, and Safari.[3]

HTTP server push

HTTP server push (also known as HTTP streaming) is a mechanism for sending unsolicited (asynchronous) data from a web server to a web browser. HTTP server push can be achieved through any of several mechanisms.

As a part of HTML5 the WebSocket API allows a web server and client to communicate over a full-duplex TCP connection.

Generally the web server does not terminate a connection after response data has been served to a client. The web server leaves the connection open so that if an event occurs (for example, a change in internal data which needs to be reported to one or multiple clients), it can be sent out immediately; otherwise, the event would have to be queued until the client's next request is received. Most web servers offer this functionality via CGI (e.g., Non-Parsed Headers scripts on Apache HTTP Server). The underlying mechanism for this approach is chunked transfer encoding.

Another mechanism is related to a special MIME type called `multipart/x-mixed-replace`, which was introduced by Netscape in 1995. Web browsers interpret this as a document changing whenever the server feels like pushing a new version to the client.[4] It is still supported by Firefox, Opera, and Safari today, but it is ignored by Internet Explorer.[5] It can be applied to HTML documents, and also for streaming images in webcam applications.

The WHATWG Web Applications 1.0 proposal[6] includes a mechanism to push content to the client. On September 1, 2006, the Opera web browser implemented this new experimental system in a feature called "Server-Sent Events".[7][8] It is now being standardized as part of HTML5.[9]

Pushlet

In this technique, the server takes advantage of persistent HTTP connections, leaving the response perpetually "open" (i.e., the server never terminates the response), effectively fooling the browser to remain in "loading" mode after the initial page load could be considered complete. The server then periodically sends snippets of JavaScript to update the content of the page, thereby achieving push capability. By using this technique, the client doesn't need Java applets or other plug-ins in order to keep an open connection to the server; the client is automatically notified about new events, pushed by the server.[10][11] One serious drawback to this method, however, is the lack of control the server has over the browser timing out; a page refresh is always necessary if a timeout occurs on the browser end.

Long polling

Long polling is itself not a true push; long polling is a variation of the traditional polling technique, but it allows emulating a push mechanism under circumstances where a real push is not possible, such as sites with security policies that require rejection of incoming HTTP/S Requests.

With long polling, the client requests information from the server exactly as in normal polling, but with the expectation the server may not respond immediately. If the server has no new information for the client when the poll is received, instead of sending an empty response, the server holds the request open and waits for response information to become available. Once it does have new information, the server immediately sends an HTTP/S response to the client, completing the open HTTP/S Request. Upon receipt of the server response, the client often immediately issues another server request. In this way the usual response latency (the time between when the information first becomes available and the next client request) otherwise associated with polling clients is eliminated.[12]

For example, BOSH is a popular, long-lived HTTP technique used as a long-polling alternative to a continuous TCP connection when such a connection is difficult or impossible to employ directly (e.g., in a web browser);[13] it is also an underlying technology in the XMPP, which Apple uses for its iCloud push support.

Flash XMLSocket relays

This technique, used by Cbox and other chat applications, makes use of the XMLSocket object in a single-pixel Adobe Flash movie. Under the control of JavaScript, the client establishes a TCP connection to a unidirectional relay on the server. The relay server does not read anything from this socket; instead it immediately sends the client a unique identifier. Next, the client makes an HTTP request to the web server, including with it this identifier. The web application can then push messages addressed to the client to a local interface of the relay server, which relays them over the Flash socket. The advantage of this approach is that it appreciates the natural read-write asymmetry that is typical of many web applications, including chat, and as a consequence it offers high efficiency. Since it does not accept data on outgoing sockets, the relay server does not need to poll outgoing TCP connections *at all*, making it possible to hold open tens of thousands of concurrent connections. In this model, the limit to scale is the TCP stack of the underlying server operating system.

Reliable Group Data Delivery (RGDD)

In services such as Cloud Computing, to increase reliability and availability of data, it is usually pushed (replicated) to several machines. For example, the Hadoop Distributed File System (HDFS) makes 2 extra copies of any object stored. RGDD focuses on efficiently casting an object from one location to many while saving bandwidth by sending minimal number of copies (only one in the best case) of the object over any link across the network. For example, Datacast [14] is a scheme for delivery to many nodes inside datacenters that relies on regular and structured topologies and DCCast [15] is a similar approach for delivery across datacenters.

Push notification

A push notification is a message that is "pushed" from backend server or application to user interface, e.g. (but not limited to) mobile applications and desktop applications. Push notifications were first introduced by Apple in 2009.[16] In 2010 Google released its own service, Google Cloud to Device Messaging. (It has since been replaced by Google Cloud Messaging and then Firebase Cloud Messaging.)[17] November 2015, Microsoft announced that the Windows Notification Service would be expanded to make use of the Universal Windows Platform architecture, allowing for push data to be sent to Windows 10, Windows 10 Mobile, Xbox, as well as other supported platforms using universal API calls and POST requests.[18]

Push notifications are mainly divided into 2 approaches, local notifications and remote notifications.[19] For local notifications, the application schedules the notification with the local device's OS, or, alternatively, sets as a timer in the application itself if it is able to continuously run in the background. When the event's scheduled time is reached, or the event's programmed condition is met, the message is displayed in the application's user interface.

Remote notifications are handled by a remote server. Under this scenario, the client application needs to be registered on the server with a unique key (e.g., a UUID). The server then fires the message against the unique key to deliver the message to the client application via an agreed client/server protocol such as HTTP or XMPP and the client displays the message received. When the push notification arrives, it can transmit short notifications and messages, set badges on application icons or play alert sounds to attract user's attention.[20] Push notifications are usually used by applications to bring information to users' attention. The content of the messages can be classified in the following example categories:

- Chat messages. E.g.: messages from Facebook messenger sent by other users.[21]
- Vendor special offers. E.g.: A vendor may want to advertise their social offers to the customers.
- Event reminder. E.g.: Some application may allow the customer to create reminder or alert for a specific time.
- Subscribed topics changes. E.g.: Users may want to get updates regarding the weather in their location, or monitor a web page to track changes, for instance.

See also

- BlazeDS
- BOSH
- Channel Definition Format
- Client–server model
- Comet
- File transfer
- GraniteDS
- Lightstreamer
- Pull technology
- Push Access Protocol
- Push e-mail
- HTTP/2
- SQL Server Notification Services
- Streaming media
- WebSocket

References

1. M. Thomson, E. Damaggio and B. Raymor (October 22, 2016). "Generic Event Delivery Using HTTP Push" (https://tools.ietf.org/html/draft-ietf-webpush-protocol-12). *Internet Draft*. Internet Engineering Task Force. Retrieved October 28, 2016.
2. "Web Notifications" (https://www.w3.org/TR/notifications/).
3. "Web Push API" (https://www.w3.org/TR/push-api/).
4. CGI Programming on the World Wide Web (http://www.oreilly.com/openbook/cgi/ch06_06.html) O'Reilly book explaining how to use Netscape server-push
5. Server-Push Documents (HTML & XHTML: The Definitive Guide) (http://victor.transformadora.com/Oreilly/wdesign/xhtml/ch13_03.htm) Archived (https://web.archive.org/web/20080417004248/http://victor.transformadora.com/Oreilly/wdesign/xhtml/ch13_03.htm) 2008-04-17 at the Wayback Machine. O'Reilly book explaining server-push
6. "Web Applications 1.0 specification" (http://www.whatwg.org/specs/web-apps/current-work/#scs-server-sent).
7. "Event Streaming to Web Browsers" (http://my.opera.com/WebApplications/blog/show.dml/438711). 2006-09-01. Retrieved 2007-03-23.

8. "Opera takes the lead with AJAX support among browsers: More efficient streaming" (https://web.archive.org/web/20070318013327/http://operawatch.com/news/2006/09/opera-takes-the-lead-with-ajax-support-among-browsers-more-efficient-streaming.html). 2006-09-01. Archived from the original (http://operawatch.com/news/2006/09/opera-takes-the-lead-with-ajax-support-among-browsers-more-efficient-streaming.html) on 2007-03-18. Retrieved 2007-03-23.
9. Server-Sent Events (https://www.w3.org/TR/eventsource/)
10. Pushlets introduction (http://www.pushlets.com/doc/whitepaper-toc.html)
11. JavaWorld article about pushlets (http://www.javaworld.com/jw-03-2000/jw-03-pushlet.html)
12. "RFC6202 - Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP" (https://tools.ietf.org/html/rfc6202#section-2.1). Retrieved 2016-05-14.
13. "XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH)" (http://xmpp.org/extensions/xep-0124.html#intro). Retrieved 2012-06-26.
14. C. Guo; et al. (November 1, 2012). "Datacast: A Scalable and Efficient Reliable Group Data Delivery Service For Data Centers" (https://www.microsoft.com/en-us/research/publication/datacast-a-scalable-and-efficient-reliable-group-data-delivery-service-for-data-centers-2/). ACM. Retrieved Jun 6, 2017.
15. M. Noormohammadpour; et al. (July 10, 2017). "DCCast: Efficient Point to Multipoint Transfers Across Datacenters" (https://www.researchgate.net/publication/316921061_DCCast_Efficient_Point_to_Multipoint_Transfers_Across_Datacenters). USENIX. Retrieved Jun 6, 2017.
16. "iPhone push notification service for devs announced" (https://www.engadget.com/2008/06/09/iphone-push-notification-service-for-devs-announced/). *Engadget*. Retrieved 2016-10-18.
17. "Google Cloud Messaging for Android (GCM) Unveiled, to Replace C2DM Framework" (https://www.infoq.com/news/2012/08/GoogleCMReplacesC2Dm). *InfoQ*. Retrieved 2016-10-18.
18. mijacobs. "Windows Push Notification Services (WNS) overview" (https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/tiles-and-notifications-windows-push-notification-services--wns--overview). *docs.microsoft.com*. Retrieved 2017-10-20.
19. "Local and Remote Notifications in Depth" (https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/WhatAreRemoteNotif.html). *developer.apple.com*. Retrieved 2016-10-18.
20. "Android and iOS Push Notifications – Blog – JatApp" (http://jatapp.com/blog/android-ios-push-notifications/). *jatapp.com*. Retrieved 2017-10-20.
21. "How do I adjust my mobile push notifications from Facebook? | Facebook Help Center | Facebook" (https://www.facebook.com/help/103859036372845?helpref=faq_content). *www.facebook.com*. Retrieved 2016-10-18.

External links

- W3C Push Workshop (http://www.w3.org/Architecture/9709_Workshop/). A 1997 workshop that discussed push technology and some early examples thereof
- HTTP Streaming with Ajax (https://web.archive.org/web/20051201013553/http://ajaxpatterns.org/HTTP_Streaming) A description of HTTP Streaming from the Ajax Patterns website
- The WebSocket API (http://www.w3.org/TR/websockets/) candidate recommendation
- HTML5 Server-Sent Events (http://www.w3.org/TR/eventsource/) draft specification