

THICK CLIENT WEB APPLICATION PROTOTYPE

Dr. K.P.Kaliyamurthie

Centre of Excellence for Big Data and Smart Cloud, BIST, BIHER, Bharath University, India.

Email: Kaliyamurthie.cse@bharathuniv.ac.in

Abstract: A thick client web application prototype is a web-application for presentation similar to that of an office application like Microsoft PowerPoint, Google Docs etc., The project introduces a framework that provides a prototype for developing modern web applications and helps in overcoming challenges involved and empower them with latest technologies like server push-pull, internet polling techniques.

1. Introduction

Page-centric Web development structures an application into individual scripts, each conscientious for processing a user request and producing a response. [1-7]Links pass control from one script to the next. This forces a go-to hardwiring of the control stream because each page must comprehend what comes next. [8-12]In spite of the software developers have long considered go-to statements destructive, they are still present in today's mainstream architecture, and they obstruct the recycle of pages in diverse parts mainstream of the application. Nowadays, numerous complex applications are fabricated with a web browser as their main client interface.

In this project we introduce, a web application system that gives a uniform and pure object-oriented perspective on web applications. We provide a prototype that could be referenced in future for creating advanced web applications.

2. Objective

This project aims at overcoming issues involved in developing modern web applications and additionally extends them with modern features. The project also intend at making the application to possess same look and feel as of a desktop application and empower them with cross platform compatibility across devices.[13-18]

This prototype is potentially capable of running in various platforms, as it is a traditional web application that just runs inside a browser sandboxed environment. Since all major browsers follow W3C web standards, the application performance and rendering is same across all

platforms and devices. Additionally the application can switch between Thick Client and Thin Client architecture so that the end users can work either offline or online which means they are independent of network availability. When network is available the application can pull the users content from the server, otherwise can make use of local file system. In case of pushing data, the same methodology follows. It introduces an abstraction layer over the asynchronous interaction protocol between the client and server to provide the illusion of using desktop application.[19-21]

This prototype also overcomes the current web application development challenges which includes hot-debugging/compilation, object re-usage, etc.,. Since the application follows web application standards and can run in server side, it is capable of serving as SaaS application in cloud environment also.[22-24]

3. Related Works

Seaside is open source and in prolific use in many business-related applications Seaside introduces a layer of abstraction over the non concurrent interaction protocol between the client and server to grant the illusion of developing a desktop application. This high level of deliberation is made possible by the reflective abilities of Smalltalk. The drawback of existing system as follows :

- Memory efficiency is a questionable factor
- Complete workload on individual client
- Network capabilities and features are not considered into account
- Lack of SaaS capabilities and advanced network features.

4. Proposed System

The proposed design composed of a "background level" written in python running in a single thread and HTML for event handling and presenting a view to the user.

All the UI components are created and reused with the single page and memory optimization is done by pushing all unfocused (but used) components to a cache file. Hot debugging and recompilation is being achieved by integrating the application with the browser add-on, which has options to debug both gui components as well as variables in the program.

Recompilation can be done on the fly without running the whole program after some modification.[25-28]

Workload on individual system is being reduced with the availability of internet connectivity. The background python thread automatically detects the availability of network and connects to the server and hereafter uses the server resources and server space rather than eating client resources.[29-31]

The Real-time sharing features allow multiple users to collaborate and run the application using a polling technique, where set of users subscribe to a single publisher and keep polling for new data and updates. The polling module keeps checking the publishers state through a central server and updating the changes on client side if there occurs a update on latter. Similarly in case of publisher the module keeps track of current publisher state to which all clients are subscribed to.

5. Overall Architecture

The basic architecture of our prototype is a three tier layer consists of presentation, processing and storage layers. The architecture consists of three layers, **Top level presentation layer** written in css, javascript and html which is responsible of displaying contents and also to receive inputs form the end user and record all events taking place at client side.

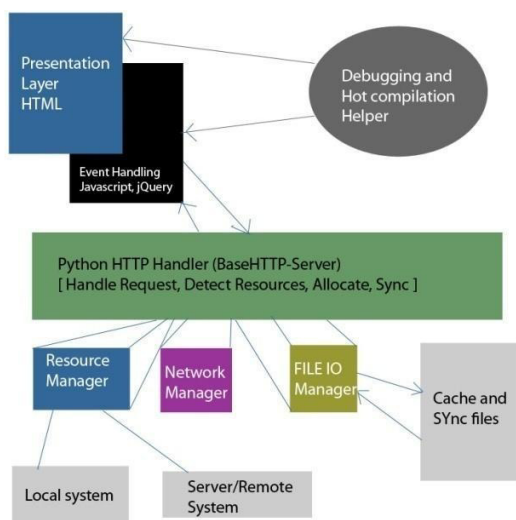


Figure 1. Architecture diagram

Middle level processing layer is a python http handler or a light weight python http server that takes responsible for processing HTTP get and post requests originating from the client side and taking the appropriate actions and responding the client with associated data/actions .this layer also takes care of resource detection and allocation, which plays vital role in reducing the workload on client. This layer is also responsible for publishing user events occurring at client side and also to subscribe to any hosts if available.

Bottom level consists of a storage layer which is primarily responsible for data storage and session state storage. The python layer decides whether the storage must be switched to local file system or remote file system based on availability of network resources.

Objects prototype Everything in the system is considered to be objects. Every objects such as image, video etc., possess of various attributes like unique_id, dimensions, appearance properties and is linked with events like click, mouse-over, mouse-out etc..

6. System Modules

a. UI and Event management module

This module is used to handle user events and inputs. Since end-users interact with the system through graphical user interface and these actions must be recorded and has to be sent to processing layer for processing.

In addition to that, performing asynchronous communication with background layer (Python thread) is also being carried out by this module.

b. Local File-IO management module

IO management module is responsible for saving and loading data to and from the local filesystem and performs reading and writing operations and maintains cache files and validates them with timestamp and last modified time.

Since the memory efficiency is improved by maintaining cache files this module decides how much data has to be cached and other attributes of cache files.

6.3 Background python thread: http handler

This module is basically a python background thread used to process all inputs originating from the front-end and to provide response to the top presentation layer. In addition to processing user events it has to take care of other tasks too. It contains following sub modules.

6.4 Caching modules

Its primary responsibility is to cache reusable objects. When an object is being reused again and again it is

being cached so as to achieve memory efficiency and fast access. When the application is closed, the state of working files and application is also saved into the cache file for later access.

6.4.1 Resource detection modules

To detect various resources available in local file system and if network is available, detect resources available in the remote environment.

Unless the external resource is available this module makes arrangements to use the current local system resources and storage features, but as soon as network availability is sensed the resources available in network is being listed and the control is transferred to the server filesystem and thus reducing the workload on client.

6.5 Network resource management module

6.5.1 Polling module

Real time sharing feature is implemented using polling module, which is used to broadcast states of users across all listening clients. The publisher constantly publishes his state and the subscribers who all subscribed to the publishers content keeps polling the change of state or actions performed by the publisher and the polling module implements this functionality. Precisely at subscriber side the module is responsible to periodically check the subscriber state and at the publisher side to update and broadcast the change of state to all subscribers otherwise called listeners.

6.5.2 Server-runtime module

Server runtime modules processes all requests from client with server runtime and allocates server resources like memory, disk space etc., so that client resource usage is minimized.

Thus an overview of modules that makes up the whole system is explained. Each module and their respective functionalities are explained. The system is mainly composed of four major modules and various sub-modules under every module.

7. Conclusion

Thus prototype successfully helps to solve challenges involved in developing modern web applications and also extend them with latest technologies and trending web features.

Our project will act as a blueprint for building high performance future web applications that could someday replace the traditional desktop applications completely.

Compared to the advantages of our prototype, there are few disadvantages too. The prototype is designed and optimized by keeping in mind with the modern computing power and architecture and hence the performance measures could be applied to modern computing speed and memory. With older browsers and low computing resources the prototype will experience a performance hit.

References

- [1] Olivier Auverlot and Stéphane Ducasse, Construire un service Rest avec Pharo et Seaside-Rest, Linux Magazine, 1, 2011.
- [2] Stéphane Ducasse, Adrian Lienhard and Lukas Renggli, Seaside a Multiple Control Flow Web Application Framework, Proceedings of 12th International Smalltalk Conference (ISC'04), 231, 257, 2004, Also Technical Report IAM-04-008, PDF.
- [3] Stéphane Ducasse, Adrian Lienhard and Lukas Renggli, Seaside: A Flexible Environment for Building Dynamic Web Applications, IEEE Software, 24, 56,63, IEEE Computer Society, 2007, 5-year ISI impact factor 2.551 (2010), PDF, DOI.
- [4] Stéphane Ducasse, Lukas Renggli, C. David Shaffer, Rick Zaccane and Michael Davies, Dynamic Web Development with Seaside, Square Bracket Associates, 2010, PDF.
- [5] Sengottuvel, P., Satishkumar, S., Dinakaran, D., Optimization of multiple characteristics of EDM parameters based on desirability approach and fuzzy modeling, Procedia Engineering, v-64, i-, pp-1069-1078, 2013.
- [6] Jayalakshmi, V., Gunasekar, N.O., Implementation of discrete PWM control scheme on Dynamic Voltage Restorer for the mitigation of voltage sag /swell, 2013 International Conference on Energy Efficient Technologies for Sustainability, ICEETS 2013, pp-1036-1040, 2013.
- [7] Kaliyamurthie, K.P., Parameswari, D., Udayakumar, R., QOS aware privacy preserving location monitoring in wireless sensor network, Indian Journal of Science and Technology, v-6, i-5, pp-4648-4652, 2013.
- [8] Sundararajan, M., Optical instrument for correlative analysis of human ECG and breathing signal, International Journal of Biomedical Engineering and Technology, v-6, i-4, pp-350-362, 2011.
- [9] Kaliyamurthie, K.P., Udayakumar, R., Parameswari, D., Mugunthan, S.N., Highly secured online voting system over network, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4831-4836, 2013.
- [10] Khanaa, V., Thooyamani, K.P., Saravanan, T., Simulation of an all optical full adder using optical

switch, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4733-4736, 2013.

[11] Raj, M.S., Saravanan, T., Srinivasan, V., A modified direct torque control of induction motor using space vector modulation technique, Middle - East Journal of Scientific Research, v-20, i-11, pp-1572-1574, 2014.

[12] Kumaravel, A., Dutta, P., Application of Pca for context selection for collaborative filtering, Middle - East Journal of Scientific Research, v-20, i-1, pp-88-93, 2014.

[13] BrinthaRajakumari, S., Nalini, C., An efficient data mining dataset preparation using aggregation in relational database, Indian Journal of Science and Technology, v-7, i-, pp-44-46, 2014.

[14] Udayakumar, R., Khanaa, V., Saravanan, T., Saritha, G., Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction, Middle - East Journal of Scientific Research, v-16, i-12, pp-1781-1785, 2013.

[15] Khanaa, V., Thooyamani, K.P., Using triangular shaped stepped impedance resonators design of compact microstrip quad-band, Middle - East Journal of Scientific Research, v-18, i-12, pp-1842-1844, 2013.

[16] Juan Lautaro Fernández, Santiago Robles, Andrés Fortier, Stéphane Ducasse, Gustavo Rossi and Silvia Gordillo, Meteoroid Towards a real MVC for the Web, Proceedings of International Workshop on Smalltalk Technologies (IWST 2009), ACM Digital Library, 2009, PDF.

[17] Web Applications wiki, http://en.wikipedia.org/wiki/Web_application

[18] PHP documentation manual, <http://php.net/manual/en/index.php>

[19] Polling techniques using PHP, http://php.about.com/od/finishedphp1/ss/simple_poll_php.htm

[20] Ullman, Chris (March 2007), Beginning Ajax, wrox, ISBN 978-0-470-10675-4, 5 July 2008 - 24 June 2008.

[21] Python HTTP server implementation, <http://docs.python.org/2/library/basehttpserver.html>

[22] Double, Chris (2005-11-05). "More on Ajax and server push". Different ways of doing server push. Revised 2008-05-05.

[23] HTML W3Schools tutorials, <http://www.w3schools.com/html/>

[24] Javascript advanced Tutorial eBook, www.tutorialspoint.com/javascript/javascript_pdf_version.htm

[25] Ajax programming wikipedia, https://en.wikipedia.org/wiki/Ajax_programming/

[26] Python documentation manual, <http://docs.python.org/>

[27] Thamothisaran, C., Prabhakar, S., Vanangamudi, S., Anbazhagan, R., Anti-lock braking system in two

wheelers, Middle - East Journal of Scientific Research, v-20, i-12, pp-2274-2278, 2014.

[28] Vanangamudi, S., Prabhakar, S., Thamothisaran, C., Anbazhagan, R., Design and fabrication of dual clutch, Middle - East Journal of Scientific Research, v-20, i-12, pp-1816-1818, 2014.

[29] Vanangamudi, S., Prabhakar, S., Thamothisaran, C., Anbazhagan, R., Design and calculation with fabrication of an aero hydraulic clutch, Middle - East Journal of Scientific Research, v-20, i-12, pp-1796-1798, 2014.

[30] Saravanan, T., Raj, M.S., Gopalakrishnan, K., VLSI based 1-D ICT processor for image coding, Middle - East Journal of Scientific Research, v-20, i-11, pp-1511-1516, 2014.

[31] Alexandre Bergel, Stéphane Ducasse and Lukas Renggli, Seaside Advanced Composition and Control Flow for Dynamic Web Applications, ERCIM News, 72, 2008, PDF.

