A kind of Cache Engine Runing in Client

Zhang Xiang
College of Information Science and Engineering
Shenyang University,
Shenyang China
E-mail: zhang__xiang@sina.cn

Liu Yang
College of Pharmaceutical Engineering
Shenyang Pharmaceutical University
Shenyang China
E-mail: nesta_viola@163.com

***Abstract*—To address the lack of intelligence and the traditional client engine server overload problem, we design a browser-based caching mechanism and ajax technology, Cache Engine (CE).It will separate the business logic from data access, and business logic will be transferred to the client implementation, while the business logic in the client-side caching of data involved, the formation of smart client system.Theory and Practice shows that CE can effectively reduce the number of server access, reduce server load and reduce network traffic, while making web applications more intelligent.**

Keywords- Cache Engine, client-side, load

I. INTRODUCTION

With the popularity of the Internet, based on B / S structure, the rapid development of applications, the role of Internet browsers is growing.But people find it provides man-machine interface and mode of operation has always been unsatisfactory.For example: poor interaction, functionality is simple, slow response, have more communications redundancy, server-side pressure, can not use client-side resources, and so on.And there is just the Ajax technology can make up for these shortcomings.

In the Ajax application model, Ajax model of engine as the core of its design and implementation of the system is directly related to the entire Web application success and failure.So how to design and implement a good Ajax Ajax application engine to become an important object of study.

Although traditional Ajax engine [1] uses asynchronous mode to communicate with more rapid response capability, but while it makes Web applications more dynamic, it doesn't reduce the server load. Although Ajax engine [2] based on RPC mode reduces server load to a certain extent, as there is no client cache, it can't reduce the access frequency of data persistence layer, and can't have the offline operate capability.

This paper presents a smart client-side Ajax technologies workflow engine-CE.It implements the traditional Ajax engine based on the use of browser cache mechanism [3], the establishment of smart client system that goal.CE is to run the business logic server to download to the browser cache implementation, and business logic involved in the data cache in the browser.When the re-run the business logic, the operation can be directly carried out in the browser cache with offline operation capability.If you need to connect the data persistence layer, CE may use "check hook" operation to access the server, update the browser cache while the "Flyweight library."To overcome the drawbacks of the traditional Ajax engine, making the Ajax engine more intelligent.

.

II. WORKING PRINCIPLE OF CE

CE design idea is to business processes to the form of XML code fragments stored on the server side, based on user requests require, by the CE dynamically load the appropriate business processes (ie, XML code fragment), followed by analysis and implementation, while the data cache. In the business process is no longer frequently used, CE will be automatically deleted from the client, while updating the cache to reduce redundant data. CE realized the logic of the business logic layer and data layer separation, it will lead to the client business logic, and server-side data is only equivalent to the logical layer provides data management and support. Based on the above analysis of CE formal definition is given below, and a drift of the XML code fragment business examples.

The following is the example of XML code fragment of flow net:

```
<Flownet>
<Entry id="dataSource">
<property name="url"> jdbc:odbc:Mysql:
//100.1.1.11:1366;DatabaseName=login
</property></Entry><Hooks>
<Hook id="login">……..</Hook>……</Hooks>
</Flownet>
```

III. FUNCTION MODULE PARTITION OF CE

A. Client component

1.Basic Function module. It includes asynchronous request device, data type converter, asynchronous data communication device, page display controller, exception handler device, etc. It is mainly responsible for initiating request, asynchronous data communication, result display, exception handler, closure of special keys and other basic functions. Basic function module loads into memory when the page starts, and resident memory is interface of CE and users.

2.Flow assembly module. It includes business flow net generator, business flow net manager, etc. which is responsible for assembly and unloading of business flow net.

Published by Atlantis Press, Paris, France.

It provides corresponding treatment schemes for request of users, and dynamically loads and unloads related business flow nets. Business flow nets are generally divided into system business flow net and user-defined business flow net. System business flow net mainly refers to those frequent applications, uninvolving interaction with servers, general business will automatically start when loading CE, usually will be president memory. User-defined business flow net is the business which is provided for some particular application service, and does not have universal, so it generally applies the way of dynamic load to manage flow assembly module. The working process of flow assembly module is divided into three steps:

① Instance of business flow net and implementation: Interpret the business logic process that business flow net defined, according to the provided entry and the corresponding parameters instance of data source, and initialize the instances of data hooks involved in business flow net, and weave data hooks into business flow net.

② Implementation navigation of the involved data hook for the process of business logic: take the related data of business logic and data hook defined of business flow net as the basis for promoting the implementation of business logic, and decide the logic trend of subsequent business according to the obtained data.

③ Complete the interaction with shareware library, and maintain data of data hook. Flow assembly module can not directly call hook data of data persistence layer, which uses the data hook to complete data hook module.

3.Data hook module. It includes data hook manager. It is mainly responsible for hook data to server, and life cycle management of data hook instance. Data hook uses data stub hook to hook remote data through remote method and invocation method.

4.Cache management module. It includes shareware manager, and is primarily responsible for shareware library management of client, including shareware validate, expired shareware update, useless shareware cancellation.

5.Controller module. It is responsible for CE load and call between modules. Controller module is equivalent to "commander" of CE, and other modules service for system through control and process of controller module finally.

B. Server component

1.Request interceptor. Analyze the request of client, and pass to request controller, and then return to the particular result of client.

2.Request controller. It is responsible for the inspection and extraction of requesting parameters, calling service, returning data.

3.Service registered warehouse. It is the hook point of data hook in service, and the interface between system and data persistence layer.

4.Data hook pool. It is responsible for the management of data hook server, is the container of data hook. Users can share data hook instance between threads by using data hook pool, which it is completely transparent for request controller. Data hook pool is created when server initializes, and cleared when stops. As data pool provides cache mechanism of data hook, it can reduce system overhead generated by frequent creation and recovery of data hook instances, save storage resources and also can make full use of the connection resources of data persistence layer.

5.Resource library of business flow net. It is responsible for providing resource download of business flow net to client.

IV. WORKING PROCESS OF CE

A. System initialization

System initialization includes two aspects, one is data preparation of server, and the other is loading of client CE. Server is the place only for providing data, which prepares data when server container initializes. It contains three aspects: First, initialize all business flow net resource by Resource library of business flow net for initiating requesting and downloading of client; second, register all data collection service by service registry pool for hooking data request of client; third, create data hook pool. CE load appears at the first request of client, and each module of CE will be loaded into the client in order.

B. User response treatment

After finishing the page load, users begins to do various operations on the page, first, the asynchronous request of basic function module captures users' operation, and submits the associated data to controller for logic processing.

C. Logic processing

Logic processing can be divided into 4 steps:

① Call business flow net manager of flow assembly module for assembly of business flow net (in case of need).

② Weave business flow net of assembly by using business flow net generator.

③ After weaving operations the flow assembly module gives hook task for data collection processing module after the current nets according to the involved hook of business flow net.

④ Controller modules continue the logic processing of other business flow net, and needn't wait for the executing results of task orders.

D. Data processing

When data collection processing module receives the hook task issued by flow assembly module, it calls the related data hook stub to hook data, hook process can be divided into:

① Generate Key. Abstract of data hook stub and its associated parameters into key.

② Local shareware hook Library. Match keys in shareware library of local client. If exists, then directly return to shareware.

③ When shareware library has no match shareware, it will call data hook to do remote hook data shareware. The process of remote hook sent to server by using http protocol asynchronous. As the application of asynchronous mode,

there may be multiple http links at the same time. That is, there are a lot of data hook instance at the same time.

④After sending remote hook request, data hook manager need to monitor the communication situation of each data hook, manage it, and wait for server return to data shareware.

⑤ After request interceptor of server receiving hook request, forward it to request controller to analyze data hook, and call the corresponding service of service registration pool according to the hook function descriptive information, and then return the executing results to the data hook.

⑥ After successfully hooking the data hook, data collection processing module will make result and generate key to form new shareware two-tuples.

⑦Call shareware manager of cache management module to validate shareware of shareware two-tuples, including cancellation shareware, update shareware, delete expired shareware, etc.

E. Conversion processing of data type

After controller module receiving the shareware two-tuples returned by data collection processing module shareware, it will submit it to data type converter of basic function module, which is responsible for converting shareware two-tuples to the recognizable format of controller module.

F. Processing of receiving data

Controller module will forward the final data to page display controller, and display the results on the page. If a process has an exception, the controller module will give it to exception handler to deal with, and forward the processing results to page display controller, and display to users.

V. CONCLUSION

Traditional model and the Ajax Web application model is the most important difference in the Ajax application model adds a Ajax client-side engine. The Ajax engine improved this paper, to put on a smart "outer"-CE. The intelligent "outer" connected to the system as the user downloaded to the middle tier running on the client browser. CE The idea is to lead the business logic layer, while the server is only equivalent to the data management center to provide data to support the back. Was put on "outer" and Ajax engine has the following advantages:

Full use of client resources. Using client-side business logic code and data reside, can maximize the use of client software resources.

Reduce the number of visits. Using the client library stores the number of shared data element, it will greatly reduce the number of visits to the server.

Reduce network traffic. Some operations must be done by the service side, particularly on the need to access the data persistence layer operations, such operations for the local refresh using ajax technology to reduce network traffic.

Reduce server load. Move the business logic processing client, reducing server load.

Make web applications more intelligent. CE can dynamically load, unload the business net, to reduce redundant data.

Has the ability to operate offline. Because CE loaded on the client, even if no interaction with the server side, CE can also use some of the local Flyweight library to provide a normal response to the request.

Improve system performance and ease of use. CE is the operating carrier, but also the user interface. The number of users is indefinite, and server performance is certain, CE idea is to change the deployment of the software structure of the original in the server's business logic and data transfer operations to the client, this can cost the same hardware The premise, improve system performance and ease of use.

However, establishment of CE in the client cache will increase the burden on the client, while the process for the number of visits and less access to web applications, CE not improve web application performance. And when the data persistence layer changes are frequent, it will have a lasting layer of data and inconsistencies Flyweight library, how flexible and effective solution to this problem requires further work.

In summary, CE applies to the following conditions: First, the system business logic and data access relatively fixed; Second, persistence layer data update less frequently; three visits a large number of process systems.

REFERENCES

[1] You Lizhen, Guo Yuchun, Li Chunxi. Principle and Application of Ajax Engine [J], Control &Automation, 2006, 20 (06) 73-75.

[2] Wang Qiang, Huang Lijuan, Yu Guoping. DWR Application in Struts Framework [J], Control &Automation, 2008, 24 (15):234-235.

[3] Wang Shengping, Li Qing. The Realization of Shared Cache Data Flow in Data Replication Management [J], Computer Engineering, 2007, 33 (20):83-85.

[4] Chen Shaoying, Liu Jianhua, Jin Chengji. Actual Combat of Load Runner Performance Test, Electronic Industry Press, 2007