GeoWeb Content Caching – Technology for Optimization and Acceleration Services

František Klímek[1]

[1]Institute of geoinformatics, VSB-TU of Ostrava, 17. listopadu 15,
708 33, Ostrava – Poruba, Czech Republic
frantisek.klimek.hgf@vsb.cz

Abstract. Goals of the research project "Orchestration services for GeoWeb" GA 205/07/0797 solved at the Institute of Geoinformatics VSB-TU Ostrava, are deal with the possibility of orchestration of web services in the field of Geoinformatics, confirming the practical possibility of the current languages for describing and planning business processes. Furthermore, is about as far as speed up the process of the evaluation of existing orchestras. Thanks to the optimization can be as a result of reduced operating costs of such orchestras. One of the possibilities that this goal could lead is to use some caching algorithms. This paper summarizes the theoretical basis and how to present web caches working with GeoWeb services content. Furthermore describe how to achieve caching services by existing web cache and describe new type of cache so-called "GeoWeb Cache" which is technology for optimization data stream and acceleration services used in GeoWeb sphere.

Keywords: GeoWeb, cache, geoinformatics, web services, orchestration.

Název: Kešovaní obsahu GeoWebu – technologie pro optimalizaci a akceleraci služeb

Abstrakt. Abstrakt. V souvislosti s výzkumným projektem „Orchestrace služeb pro GeoWeb" GA 205/07/0797 řešeného na Institutu geoinformatiky VŠB-TU Ostrava, zabývajícího se možností orchestrace webových služeb z oblasti geoinformatiky a ověřením praktických možností dostupných jazyků pro popis a plánování obchodních procesů, vyvstala i myšlenka co nejvíce urychlit provádění požadovaných orchestrů služeb, ušetřit objem dat přenesených sítí a výsledně taktéž zlevnit provedení žádaných orchestrů. Jednou z možností, která by k tomuto cíli mohla vést je využití kešování. Příspěvek popisuje, jak současné webové keše přistupují k obsahu GeoWebových služeb a zda dochází k jeho kešování. Zmiňuje možné způsoby, jak docílit kešování obsahu služeb tradičními kešemi a popisuje návrh nového druhu tzv. "GeoWebové keše", která by napomohla optimalizovat datové toky v oblasti GeoWebu.

Klíčová slova: GeoWeb, keše, geoinformatika, webové služby, orchestrace.

1 Introduction

In recent decades can be traced rapid development of Internet and technology related to him. Internet, which was originally used only for the sharing of information and research results, with over time become a very powerful tool that throughout all sectors of human action.

The Internet is used not only for easy sharing and linking texts, multimedia presentations, streaming content, but even as an arena for distributed applications.

With referred growing range of use of the Internet is also expanding and amending the requirements for caching its contents. From the original caching website, which have been largely static in the past - changing its content rarely are moved to the dynamically generated web site, where it is already having caching their content more complex. Very frequently, however, and now uses caching pictures and files that basically does not change their content. With approach of multimedia, streaming music and videos have emerged again, the new requirements to caching this specific content.

The next step, and now the great trend of the use of the Internet, not only in the field of information technology, but also geoinformatics, which is usually based on these trends, are distributed applications, arranged in the Web services and components free or commercially available within the global network. At present, the quantity of services providing content to geoinformatics region are increasing.

As part of the research project "Orchestration services for GeoWeb", dealing with the possibility of orchestration of web services in the field of Geoinformatics, arise the idea as far as possible to accelerate the implementation of the orchestras services, to save the data carried over networks and result also cheapen the implementation of the claimed orchestras.

One option, which would contribute to this goal, may be use of caching. How to Web cache access to content of GeoWeb services? Is there any way to take advantage of the current cache to accelerate the implementation of orchestras? What are the requirements for cache, which could accelerate the implementation of orchestras?

2 GeoWeb services Orchestration

Orchestration is one of two forms for service coordination. Service coordination concerns how to services work together to fulfill a business process [6].

Web Service Orchestration (WSO) promises the ability to integrate and assemble individual Web services into standards-based business processes. WSO is an key enabler for an enterprise's Service Oriented Architecture and a critical layer in the web services technology "stack". These loosely coupled business processes will be designed, integrated, executed and managed similar to the way proprietary Enterprise Application Integration (EAI) and Business Process Management (BPM) tools operate today. However, business process execution standards and Web services will greatly reduce vendor lock-in to dramatically reduce costs and provide broader interoperability and reusability benefits [8].

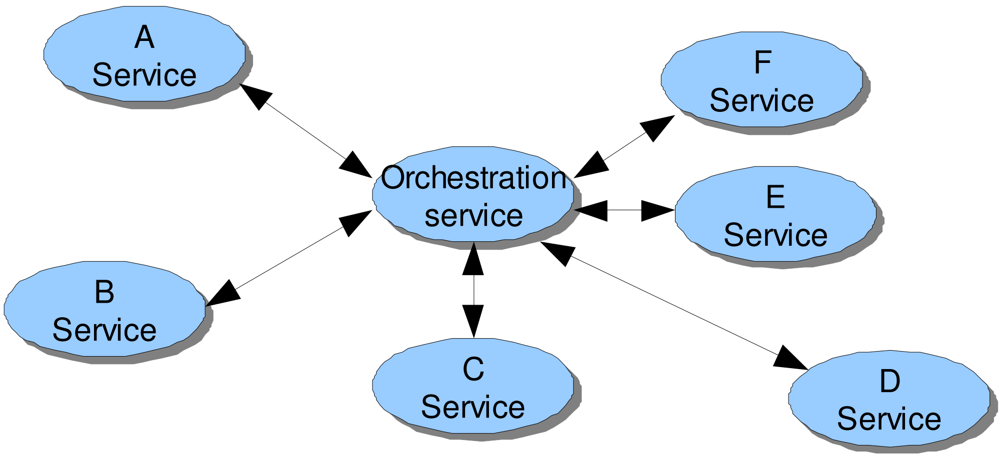

Figure 1. Orchestration [6]

As figure 1 illustrates, orchestration refers to a form of processing in with one service acts as a controlling hub in relation to other services, which act as spokes. The hub might receive a message from one spoke and make decisions based on that message, as by changing the format or content of the data and invoking some other spoke.[6]

Learn more about GeoWeb services orchestration, as addressed in the context of a research project "Orchestration services for GeoWeb" GA 205/07/0797 can be found in article "Do we know how to orchestrate on the GeoWeb platform?", which is part of this proceedings.

As already mentioned orchestration allows to link services to the functional units. Imagine for example, a daily run orchestra, which analyzes meteorology situation. This orchestra is based on the actual rainfall, the level of rivers, and other data rendered the state and forecast threats. For visualization will be used orthophoto data from WMS servers, which are updated once every 3 years. Furthermore, WMS data showing the boundaries of municipalities and cover, which are updated once a year. In addition to the orchestra will be involved in road data presented by WFS service. The data is used to create overlap, and the designation of the road, where flooding may occur. Roads data are updated only once every 2 years. Therefore, if this analysis using the orchestra performed daily throughout the Moravian-Silesian region, will create unnecessary questioning to origin server providing this data. Questions and answers will remain the same and the servers are unnecessary exploitation. Would therefore not be appropriate to take advantage of caching the content of these GeoWeb services for the purpose of speeding up and reducing the load on the original server?

This article, however, will not only address the cache, which would be suitable only for the needs of Orchestration, but looks to cache generally, if at all caching of GeoWeb services. Describe the background and possible proposals for the caching of GeoWeb services.

2.1 GeoWeb services

The Open Geospatial Consortium (OGC) is non profit, international, voluntary consensus standarts organization specializing in geodata and WebServices. The OGC consist of over 300 organizations from government, academia, industry and others. The OGC was founded on the concept of providing open specifications at no cost to the public to acquire or implement. The OGC also leverages existing efforts from other standards organizations such as the W3C, ISO and OASIS [5].

One of the main activities of the OGC Web Services testbed is captured in an Interoperability Program Report (IPR) that describes a common architectural framework for web-based geospatial services. This framework specifies the scope, objectives and behavior of a system and its functional components which are common to all such services and extensible for specific services and service types. To the extent that this is also reference architecture in the spirit of ISO 19119, it should be independent of particular technology choices [9].

OpenGIS Web Services (OWS) are individual components of dynamic geospatial computing applications; they are also parts of an overall paradigm for building solutions to geospatial “problems”. This paradigm - the Spatial Web - imposes both conceptual and implementation constraints on how OWS works. Conceptual constraints include service orientation, n-tier distribution capability, self-description, and stateless operation; these constraints generally address functionality [9].

The best known and currently also most used services are following (The following lines in this chapter are based on these OGC documents [9], [10], [11], [12].):

Web Map Service (WMS) – produces maps of spatially referenced data dynamically from geographic information. This International Standard defines a “map” to be a portrayal of geographic information as a digital image file suitable for display on a computer screen. A map is not the data itself. WMS-produced maps are generally rendered in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats.

Web Feature Service - The OpenGIS Web Feature Service Interface Standard (WFS) defines an interface for specifying requests for retrieving geographic features across the Web using platform-independent calls. The WFS standard defines interfaces and operations for data access and manipulation on a set of geographic features, including:

- Get or Query features based on spatial and non-spatial constraints
- Create a new feature instance
- Get a description of the properties of features
- Delete a feature instance
- Update a feature instance
- Lock a feature instance

Among the other interesting and in the subconscious receiving the standards include, for example, Web Coverage Service (WCS) which defines a standard interface and operations that enables interoperable access to geospatial "coverages" such as satellite images, digital aerial photos, digital elevation data, and other phenomena represented by values at each measurement point, or Web Processing service (WPS) which defines how a client can request the execution of a process, and how the output from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients’ discovery of and binding to those processes. The data required by the WPS can be delivered across a network or they can be available at the server.

At present, the only distributed computing platform (DCP) explicitly supported by OGC Web Services is the World Wide Web itself, or more specifically, Internet hosts implementing the Hypertext Transfer Protocol (HTTP). Thus an HTTP Uniform Resource Locator (URL) locates the Online Resource of each operation supported by a service instance. The URL may be different for each operation, or the same, at the discretion of the service provider.

HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular web feature service and offered by a service instance. The use of the Online Resource URL differs in each case.

An Online Resource URL intended for HTTP GET requests, is, in fact, only a URL prefix to which additional parameters must be appended in order to construct a valid Operation request. A URL prefix is defined as an opaque string including the protocol, hostname, optional port number, path, a question mark '?', and, optionally, one or more server-specific parameters ending in an ampersand '&'. The prefix uniquely identifies the particular service instance. A client appends the necessary request parameters as name/value pairs in the form "name=value&". The resulting URL must be valid according to the HTTP Common Gateway Interface (CGI) standard, which mandates the presence of '?' before the sequence of query parameters and the '&' between each parameter. As with all CGI applications, the query URL is encoded to protect special characters.

The URL prefix must end in either a '?' (in the absence of additional server-specific parameters) or a '&'. In practice, however, Clients should be prepared to add a necessary trailing '?' or '&' before appending the Operation parameters defined in this specification in order to construct a valid request URL.

An Online Resource URL intended for HTTP POST requests is a complete and valid URL to which Clients transmit encoded requests in the body of the POST document. A WFS must not require additional parameters to be appended to the URL in order to construct a valid target for the Operation request.

In addition to or instead of offering web feature services using the HTTP protocol, a service provider may offer web feature services using HTTPS. HTTPS is HTTP over a secure communication channel which allows encrypted information to be transferred between machines over the World Wide Web. The use of HTTPS does not affect the description of the requests and responses described in this specification but may require additional actions to be taken on both the client and the service in order to initiate the secure communication. The description of those actions is beyond the scope of the this specification.

The services mentioned are used for encoding one of the following methods. The first uses XML as the encoding language, and is intended to be used with HTTP POST method. The second encoding uses keyword-value pairs (KVP) to encode the various parameters of a request and is intended to be used with HTTP GET.

When using the HTTP POST method, the content type for XML encoded WFS requests must be set to text/xml. When using the HTTP POST method, the content type for KVP encoded WFS requests must be set to application/x-www-form-urlencoded and the content of the document must be equivalent to the query string of an HTTP GET request. That is, the content must be equivalent to the string that follows the ‘?’ character in a URL encoded GET request. Of course, the content must be encoded to protect special characters.

When using the HTTP GET method and KVP encoded WFS requests, a MIME type is not applicable as the entire request is encoded in the URL as keyword-value pairs that follow the ‘?’ character. The combination of XML encoded requests and the HTTP GET method is not supported.

For communication between a WFS, WCS, WPS client and and services using HTTP POST method can be used SOAP messages.

The Simple Object Access Protocol (SOAP) is a communication protocol for communication between applications. It defines a format for sending messages between communicating applications via the Internet and specifically using HTTP. Soap is platform independent, language independent and SOAP messages are encoded using XML. This means that SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

3 Cache

The term cache has French roots and means, literally, to store. As a data processing term, caching refers to the storage of recently retrieved computer information for future reference. The stored information may or not be used again, so caches are beneficial only when the cost of storing the information is less than the cost of retrieving or computing the information again [18].

3.1 Web cache

A Web cache sits between Web servers (or origin servers) and a client or many clients, and watches requests for HTML pages, images and files (collectively known as objects) come by, saving a copy for itself. Then, if there is another request for the same object, it will use the copy that it has, instead of asking the origin server for it again [2].

There are two main reasons that Web caches are used:

- To reduce latency - Because the request is satisfied from the cache (which is closer to the client) instead of the origin server, it takes less time for the client to get the object and display it. This makes Web sites seem more responsive.
- To reduce traffic - Because each object is only gotten from the server once, it reduces the amount of bandwidth used by a client. This saves money if the client is paying by traffic, and keeps their bandwidth requirements lower and more manageable [2].

The main concept is based on the fact that all caches have a set of rules that they use to determine when to serve an object from the cache, if it's available [2].

3.2 Web cache and GeoWeb content

In connection with the OGC Web services it is necessary to mention the following issues that caching any of these services are complicated. The following facts, relating to the problem arising in the context of the current web-cache.

- Blank header

Current Web caching systems are designed especially for website caching and based on facts and indications related to the HTTP protocol, whether in version 1.0 or 1.1. We have noted that these Web caches are incooperative with the output of mentioned services. This is basically a similar situation as in the case of output generated by using scripting (eg CGI scripts). Web cache are based primarily on data provided through the HTML meta tags - designed more for the client cache or via HTTP headers - in particular those allowing caching at a higher level.

In the case of the use of GET method and KVP encoding should be the current Web cache enough, theoretically. Communications are done via the HTTP protocol and should not in the content caching a problem.

When looking at returning the response headers of these services and their contents will be seen that all emptiness. That means that the current Web cache, usually defined by the rules, the content of these services no caching, because do not find any information on whether to be or not to perform. Caches therefore have no input arguments for the evaluation according to their internal decision-making rules. It is necessary, headers generating and their content is in arrangements of the original servers, thus providing services. This means, therefore, that for this fact alone may be the original servers.

- Low probability of repeatability of responses

Next problem is not too frequent repetition of the same requirements, meant exactly the same URL requests. Looking to OGC specifications at many of parameters which can, including their values to enter the URL request is only a small number of cases likely to be repeated the same question. Just for example, noted that the shift of shunting display area is changed with each step value bbox and thus changed the URL of the query. The diversity of these questions, ineffective and cause caching content of these services, because there is a need constant questioning on the original server and Web server cache rather then the communication has slowed down.

- The problem with POST method

Quite specific problem will occur for calls using the POST method. The POST method is by current caches (practically) impossible to cache; if you send information in the path or query (via GET), caches can store that information for the future. POST, on the other hand, is good for sending large amount of information to the server (which is why it won't be cached; it's very unlikely that the same exact POST will be made twice) [7]. With POST method informations are sent in the body of the query, so is not visible in URL. For this reason, the caching content services, which are used to send a query POST method, there are no caching.

4 Proposal for a solution to the GeoWeb content caching problem

The problem can be resolved in two understanding planes. The first plane is to use current Web cache without any changes, without modifications. The second plane solution is to focus on the Web cache, and you adjust to the shape to be able to work with a GeoWeb services. Each of the nominated solution has its own specifics, which are summarized below.

To resolve the problem at level based on current Web cache, would therefore have to be to modify the content presented by GeoWeb services in the state of the current Web cache able to work with the contents of this services. This way useful for presenting data services especially, such as WMS and WFS, is the possibility of completing the HTTP headers into a friendly form for current Web caches. These headers, however, had to be completed only if a lot of repeated requests. Otherwise, could create unnecessary burdens on web caches. Repetition of the same query is the unlikely but still occurring. For example, a start page of web map application, whene every time you start the homepage there is a repeated call default map. In this case, would have to be on the side of the map server implemented mechanisms for adding HTTP response headers. In a better case together with a mechanism for the analysis of frequently queries. This solution - the use of existing Web cache - it is limited to questions encoding method KVP using HTTP GET method - this is based on the design and implementation of the current Web cache. It is also noted that the efforts made to implement the suggested solution, given the relative likelihood of repeated and unrepeatable questions have been significant in the overall concept unprofitable.

The second plane solutions, as already mentioned, is aimed at Web cache, and their treatment. This solution is aimed at adjusting or creating a new kind of cache. So what's the main reason for the proposal for this new kind of caches? In particular the need to eliminate these problems. Firstly problem with the requirement of the caching of content GeoWeb services is a need to tackle is to allow the cache to work not only with the GET method of http protocol, but allow the cache to enter into the response body in the case of POST method. Second, this newly created type of cache could more understand the query, analyze it and possibly return a subset of the outcome, such as the drawing up of the outcome of previous queries. This feature is already at the border, where it is possible to indicate the cache, on the basis of functionality, such as web map server providing the service. In GeoWeb cache, therefore, there is a gradual replication of data returned to the original server. Thirdly, this cache could be implemented mechanism for analysis of often carried requests, which has already been mentioned in previous solutions built on the change of the GeoWeb services content. In the last, server responses would even be able to frequently asked questions in the pre-cache artificially. Pre-cache content could be adjusted manually using the user-friendly interface.

5 GeoWeb Cache

Geoweb Cache (GWC), this type of cache may be designed according to the levels corresponding to the location of the cache in the architecture of the network. The locations corresponding with locations along the path between a client and origin server. First, in current architecture many browsers and other user agents have build-in caches. Next, a caching proxy (a.k.a. “proxy cache”) aggregates all of the requests from a group of clients. Lastly, a surrogate can be located in front of an origin server to cache popular responses [18].

5.1 Client site solution

At present, browsers and other user agents benefit from having a built-in cache. When you press the Back button on your browser, it reads the previous page from its cache. Non graphical agents, such as web crawlers, cache objects as temporary files on disk rather than keeping them in memory [18].

Netscape Navigator lets you control exactly how much memory and disk space to use for caching, and it also allows you to flush the cache. Microsoft Internet Explorer lets you control the size of your local disk cache, but in a less flexible way. Both have controls how often cached responses should be validated. People generally use 10-100 MB of disk space for their browser cache [18].

Analogy could be GeoWeb client cache, implemented in some of the desktop GIS applications, the open-source, for example, OpenJUMP [14], QGIS [4], or some commercial application.

A browser cache is limited to just one user, or at least one user agent. Thus, it gets hits only when the user revisits a page [18].

GeoWeb client cache would be implemented only for one particular product and using only by one user may be ineffective. The user cans one and the same content in a web browser and GIS application, then it is GeoWeb cache inoperable.

In the event that the user can access the GeoWeb content from multiple applications, it is preferable to use a solution based on the proxy.

5.2 Proxy solution

Caching proxies, unlike browser caches, service many different users at once. Since many different users visit the same popular web sites, caching proxies usually have higher hit ratios than browser caches. As the number of users increases, so does the hit ratio [18].

Caching proxies are essential services for many organizations, including ISPs, corporations, and schools. They usually run on dedicated hardware, which may be an appliance of a general-purpose server, such as a UNIX or Windows NT system. Many organizations use inexpensive PC hardware that costs less than $1,000. At the other end of the spectrum, some organizations pay hundreds of thousands of dollars, or more, for high-performance solutions from one of the many caching vendors [18].

Caching proxies are normally located near network gateways (i.e., routers) on the organization's side of its Internet connection. In other words, a cache should be located to maximize the number of clients that can use it, but it should not be on the far side of a slow, congested network link [18].

Proxy sits between clients and servers. Unlike browser caches, a caching proxy alters the path of packets flowing through the network. A proxy splits a web request into two separate TCP connections, one to the client and the other on the server. Since the proxy forwards requests to origin servers, it hides the client's network address [18].

One of the most difficult aspects of operating a caching proxy is getting clients to use the service. Many organizations use interception caching to divert their network's HTTP traffic to a cache. Network administrators like interception caching because it reduces their administrative burdens and increases the number of clients using he cache. However, the technique is controversial because it violates and breaks protocols (such as TCP and HTTP) in subtle ways [18].

Not just a Web browser, as well as GIS applications is usually prepared for connection with the Internet using proxy servers and include an offer that allows you to configure the proxy. Another option is to set the proxy directly to the system variables – then any communication over the Internet runs through proxy.

Likelihood of a recurrence of the same query, therefore, increases not only the growing number of users using the cache, but also the increasing range of applications that access to the proxy cache.

Practical implementation would be realized by adjustments to any of the currently used proxy-cache server applications that are already widely deployed in organizations and used. Extend the functionality to this application. Most of currently used proxy-cache server may be extended by write a plug-in. The only editing it would be the creation of GWC as a form of plug-in.

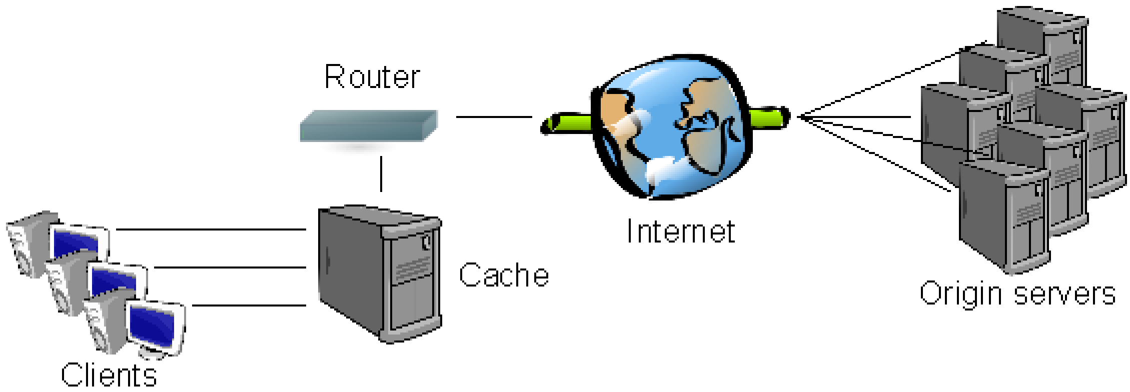

Figure 2. Proxy Cache Configuration Example [1]

5.3 Surrogate solution

Surrogate - the term used to describe reverse proxy servers, server accelerators, and other devices that pretend to be origin servers. RFC 3040 defines a surrogate:

> *A gateway co-located with an origin server, or at a different point in the network, delegated the authority to operate on behalf of, and typically working in close cooperation with, one or more origin servers. Responses are typically delivered from an internal cache.*

Surrogates are useful in number of situations. Content distribution networks use them to replicate information at many different locations. Typically, clients are directed to the nearest surrogate that has a given resource. In this manner, it seems like all users are closer to the origin sever [18].

Another common use for surrogates is to “accelerate” slow web servers. Of course, the acceleration is accomplished simply by caching the server's responses. Some web servers are slow because they generate pages dynamically. For example, they may use Java to assemble an HTML page from different components stored in a relational database. If the same page is delivered to all the clients who visit, a surrogate will accelerate the server. If the pages are customized to each client, a surrogate would not speed things up significantly [18].

Surrogates are also often used to decrypt HTTP/TLS connections. Such decryption requires a fair amount of processing power. Rather than put that burden on the origin server itself, a surrogate encrypts and decrypts the traffic. Although communication between the surrogate and the origin server is unencrypted, there is little risk of eavesdropping because the two devices are usually right next to each other [18].

Surrogates that cache origin server responses are not much different from caching proxies. It's likely that any product sold as a client-side caching proxy can also function as an origin server surrogate. It may not work automatically, however. You'll probably have to configure it specifically for surrogate operation [18].

The workload for a surrogate is generally much different from that of a caching proxy. In particular, a surrogate receives requests for a small number of origin servers, while client-side proxies typically forward requests to more than 100,000 different servers. Since the traffic arriving at the surrogate if focused on a small number of servers, the hit ratio is significantly higher. In many cases, surrogates achieve hit ratios of 90% or more [18].

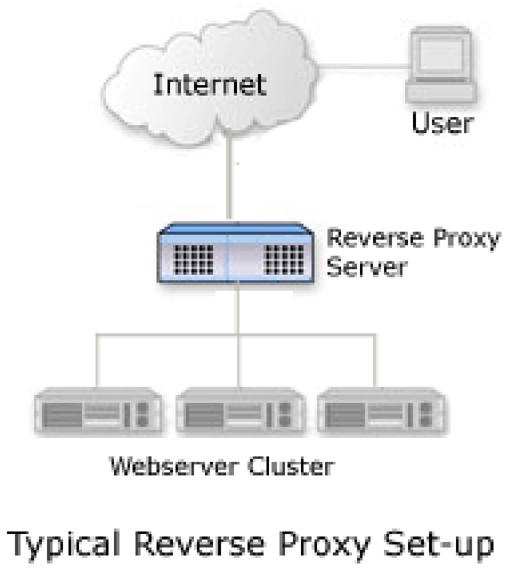

Figure 3. Reverse Proxy [17]

So in case of repeated requests reverse proxy cache don't request server providing the service, but allow the return result directly from its internal cache memory. This method is especially suitable in the case of services for which there can be a long time, the implementation of the calculation. In geoinformatics branch is this method especially suitable for WPS, or mentioned orchestras.

Server responses would even be able to frequently asked questions in the pre-cache artificially.

Other features of the reverse proxy could be possibility of completing the http headers into a friendly form for current web cache. These headers, however, had to be completed only if a lot of repeated requests. Otherwise, could create unnecessary burdens on web caches. Repetition of the same query is the unlikely but still occurring. For example, a start page of web map application, whene every time you start the homepage there is a repeated call default map. In the context of orchestration, however, there can be repeated more frequently asked questions.

For example, orchestra, which wills each week to analyze the changes in the real estate market and subsequently visualize, will be regularly used for visualization of the same data. With the update of the data occurs only once every 3 years. The original site would be about 150 times during thus treated unnecessarily. The original site would be unnecessarily over-exploitation and fell to speed its response.

The main point for both mentioned solution would therefore be proposing an appropriate mechanism that could effectively search for recurrent requests, and they subsequently implement caching, or edit the header.

5.4 Service mirroring

Last way is different from previous and specific mainly for designed way of orchestration. It is the use of substitute services, which could access the same data, but, for example, in a shorter time. This is the kind of mirroring. As part of the orchestras will be possibility of defining substitute services, which in the case of the requirements for faster implementation will be used instead of the original service. It is a creation of a separate service with its own interface, but the content will be identical to a service that will replicate.

5.5 Solutions suitable for orchestration

For providers of Web services could be another option used GeoWeb caching of content in the form of reverse proxy. For users, whether from a number of current users, administrators and local networks, it is interesting possibility GeoWeb proxy cache. In terms of orchestras, solved in the context of the grant, are appropriate options proxy cache and service mirroring.

6 Summary Conclusions

The paper is designed as a source of facts and a description of the problem arising from the requirement of GeoWeb services content caching for the purpose of accelerating the response of services. The article stated consideration of the possible ways of solution, to conclude with a focus on orchestration of GeoWeb services. The article describes a new concept - GeoWeb cache. At present, there is a more detailed proposal mentioned GeoWeb cache, and subsequently will be carried out tests and evaluation of the results of those tests so far are in the application of this proposed and implemented GeoWeb cache to accelerate services.

Next direction of the project and the proposal GeoWeb cache will seek to establish the plug-in for one of the widely-used proxy cache server. For example, is among them include SQUID [15], Apache, in the form of cache server [16], or some other [3]. Most of these applications allows the setting not only into a proxy server form, but as well as may be used as a reverse proxy cache.

Support

The article is supported by Grant agency of the Czech republic GACR as a part of the project GA 205/07/0797 GeoWeb services orchestration. The article is supported by open source community as well. We have used open source projects GeoNetwork Open Source, WSCO, Apache Tomcat, Jetty, Open Office, GIMP, Dia, PostGIS, PHP, PostgreSQL, Apache HTTP Server, GNU/Linux Ubuntu, GNU/Linux Debian, X11, MySQL, Open Clip Art Library, Freefont and others for this article.

Reference

1. Cisco. CSS Advanced Configuration Guide. Configuring Caching. Online. Resource: <http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11000series/v5.00/configuration/advanced/guide/Caching.html>

2. Davison B. D. *Web Caching*. Online. Resource: <http://www.web-caching.com/>

3. Davison B. D. *Proxy Caches*. Online. Resource: <http://www.web-caching.com/proxy-caches.html>

4. Quantum GIS home page. Online. Resource: <http://www.qgis.org/>

5. Kralidis A. T. *Geospatial Web Services: The Evolution of Geospatial Data Infrastructure*. The Geospatial Web : how geobrowsers, social software and the Web 2.0 are shaping the network society. London 2007. ISBN. 978-1-84628-826-5

6. Margolis B., Sharpe J. L. *SOA for the business developer: concepts, BPEL, and SCA*. MC Press. 2007 Lewisville. ISBN 978-1-58347-065-7.

7. Nottingham M. *Caching Tutorial for Web Authors and Webmasters*. Online. Resource: <http://www.mnot.net/cache_docs/>

8. Schittko Ch. *Web Service Orchestration with BPEL*. XML conference & Exposition 2003. [cit. 2008-10-11]. Online. Resource: <http://www.idealliance.org/papers/dx_xml03/papers/04-06-01/04-06-01.html>

9. OGC. OpenGIS Web Services Architecture. Online. Resource: <http://www.opengeospatial.org/>

10. OGC. OpenGIS Web Feature Service (WFS) Implementation Specification. Online. Resource: <http://www.opengeospatial.org/>

11. OGC. OpenGIS Web Map Service (WMS) Implementation Specification. Online. Resource: <http://www.opengeospatial.org/>

12. OGC. Web Coverage Service (WCS) Implementation Standard. Online. Resource: <http://www.opengeospatial.org/>

13. OGC. OpenGIS Web Processing Service (WPS) Interface Standard. Online. Resource: <http://www.opengeospatial.org/>

14. OpenJUMP project home page. Online. Resource: <http://openjump.org/wiki/show/HomePage>

15. Squid-cache home page. Online. Resource: <http://www.squid-cache.org/>

16. The Apache Software Foundation home page. Online. Resource: <http://apache.org/>

17. Visolve Squid Team. *Implementing Reverse Proxy Using Squid*. Online. Resource: <http://www.visolve.com/squid/whitepapers/reverseproxy.php>

18. Wessels D. *Web caching*. O'Reilly. Beijing 2001. ISBN 1-56592-536-X.