A Service Oriented Framework for Animating Big Spatiotemporal Datasets

Ahmet Sayar
Department of Computer Engineering, Kocaeli University
Izmit/Kocaeli, 41380, Turkey
ahmet.sayar@kocaeli.edu.tr

Abstract. We propose a service oriented distributed system framework for animating big spatiotemporal vector datasets. The aim is exploiting the patterns in spatial data dynamically changing over time. Animation consists of successively played and temporally related still map images. Each map image in the animation is a satellite map enriched with (or plotted over) spatiotemporal datasets. The system components are designed as web services. We also extend open standards' GIS web services definitions with topic-based publish-subscribe paradigm, which best suits to the animation requirements.

Keywords: Distributed Systems, animation, GIS, animated map, spatiotemporal data.

1 Introduction

Vast amounts of data related to earth are time-series and spatial in nature. Spatial data are preferably represented and displayed as map layers. When you have the same layer in several different moments along the time, it is better to display them as part of a movie. This is called time-series animation, which is a visualization technique ideally suited for the display and analysis of spatiotemporal and geographic data sets. Map animations enable scientific analysis and results to be understood not only by the scientist but also the public and policymakers from different domains and education levels. Animated maps can be interpreted more easily than their static representations by the users.

We propose a service oriented distributed system framework [1] for map animations. Maps are created from spatiotemporal datasets. Services in the system are defined with standard bodies (Open Geospatial Consortium (OGC) [2] and ISC-TC211). Standardization offers advantages for data sharing, combining software components and for overlaying graphical outputs from different sources. However, standardization comes with its costs. Costs mostly come from the fact that web services are based on XML based SOAP over HTTP protocol and data to be processed are z-encoded in Geographic Markup language (GML) [3] which is an XML-based format. To overcome such problems in a distributed system framework requiring large scale XML-encoded geographic feature sets, we have investigated the possibilities of using topic-based publish-subscribe paradigms (which is mostly used in P2P systems) for exchanging data payload between web services. To do so, we have integrated NaradaBrokering [4] to the communication between OGC compatible standard GIS (Geographic Information Systems) Web Services. Web Services are well-known and widely used RPC (Remote Procedure Call) systems. In basic RPCs, services (client/server) tightly coupled in terms of space and time. This is an example of synchronous communication in which client and server must be active at the same time. The proposed system has a couple of advantages over using pure Web Services. It enables simple creation from the partially returned data and gets rid of SOAP (Simple Object Access Protocol) message creation overheads. Moreover, by using NaradaBrokering (event and topic-based publish/subscribe system) we utilize network level quality of services provided in it.

After developing an efficient data transfer protocol between standard GIS [5] web services, we propose an animation web service extended from Web Map Services (WMS) [6]. WMS have capabilities of animating temporally related map images one by one dated in a vertical frame as a film. Images are created from the spatiotemporal data provided by Web Feature Services (WFS) [7].

2 Related Work

In the last decade renewed interest in animation has emerged due to technological developments. Because of these developments it is nowadays relatively easy and inexpensive to construct animations. This has led to an increase in the number, variety and complexity of animations produced. One of the areas that animations have been successfully applied is map animations.

Map animations have been studied by various disciplines such as computer sciences, remote sensing and pattern recognition. Most of the applications [8-12] are central (i.e. desktop) in which data and services are physically located in the same machine and the analyses are carried out in the same place. These early works are on recognition of temporal changes in spatial datasets through animation, but our focus is creating animating web services enabling sharing and collaboration of animated spatial data among the virtual organizations through the distributed systems.

After the invention of the internet and advancements in distributed systems there has been great dissemination possibilities, and it has become easier to access data and processing services and coupling them for the application purposes. In this context, there some client-centric and server-centric solution approaches. MathWorks's mapping toolbox [13] and GeoServer's animator toolbox [14] can be given as examples of client-centric approaches. They connect to remote WMS (OGC compatible) and fetch the map images to create an animation. They build animations as a set of frames, and each frame is a separate WMS getMap call, similar to the others in the set, with but with a different value in one of the parameters. Map images are stored into local file system at client side, and animations are created as animated GIF or movies in AVI format. Köbben [15], Becker [16] and Esri's ArcGIS [17] can be given as examples of server-centric map animation approaches, but they do not develop their services in accordance with the service oriented architectures. Furthermore, they do not consider the issues of large scale data transfer over the internet.
network for the real time animations. The animations they produce are mostly in the form of moving window (bounding box) or zooming in and out.

Compared to the related works, our solution approach is considered as server-centric. The proposed system not only supports zooming or moving animations but also animates spatiotemporal changes in large scale feature collections for a specific bounding box. To do that we take data rendering and overlaying issues into considerations, and propose a distributed system framework for collaborating and sharing the animated maps as streaming data through Real-time Transfer Protocol (RTP). This enables animations to be played in collaborative and Grid [18] environments.

For the efficiency and applicability, we adopted publish-subscribe system to the communication between OGC compatible standard GIS Web Services. This removes the deficiencies of synchronous RPC communications in GIS Web Services. We could also use some other alternative paradigms such as shared spaces approach and message queuing approach [1] for the same purposes. In shared space approach, producers insert messages asynchronously into a container, and consumers pulls (i.e., reads synchronously) from the container. In this approach services (client/server) tightly coupled in terms of space and time. In case of basic message queuing systems, producers insert messages asynchronously into FIFO and consumers reads synchronously (pull) from FIFO. This approach also provides time and space decoupling.

As you realize shared spaces and message queuing approaches are time and space decoupled. On the other hand, publish/subscribe is an event-based system and provides decoupling in terms of time, space and synchronization as well.

3 Architecture: Streaming Map Movies

This paper proposes a distributed service oriented architectural framework for binding map based geo-data animations to the distributed services by adopting and implementing OGC’s WFS and WMS services in accordance with the publicly available standards. OGC also has a discussion paper [19] on animations as an extension to WMS. It discusses how WMS specifications can be extended to allow animations that move in space over time. In this case, the only parameter changing is the bounding box in successive map images. On the other hand, we focus on overlay layers which are rendered from feature data collections represented as GML and provided by WFS. We take both geometrical and non-geometrical attributes of spatiotemporal data sets into account and animate their changing values over time. We produce map animations in the form of streaming map videos similar to ones you see in the weather cast web sites.

The work presented here looks into the possibilities of extending the web based map services with time-series data as animated maps, and provide a framework enabling integration of animation services to the distributed systems. In the proposed framework, for the service level interoperability, in terms of request and response types, and definition of animated layer descriptions we use OGC defined standards [6,7], but for the data transfer we propose a novel approach based on integration of OGC specifications with web services [18] and topic-based publish-subscribe paradigms.

3.1 Publish/Subscribe Paradigm

Publish/subscribe systems provide a useful paradigm for selective data dissemination and most of the complexity related to addressing and routing is encapsulated within the network infrastructure. Publish/subscribe systems utilize multicast networking facilities (also at data link level). It is actually a broker layer overlay network, which is based on transport level connections between nodes.

Subscribers register their interest in an event and are subsequently asynchronously notified of events generated by publishers. Publishers are generators of events and subscribers are consumers of events [20].

Publish/subscribe systems enable loosely coupled form of computer communication and interaction. Information generation and consumption is independent from each other. This feature is also called decoupling in terms of time, space and synchronization. Parties don't need to be active at the same time. Publisher can generate events when a subscriber is disconnected. Subscriber can be notified when publisher is disconnected (see Fig. 1).

![Fig. 1. Event-based Publish/Subscribe paradigm](image)

There are three types of publish/subscribe paradigm, topic-based, content-based and type-based. In a topic-based system, messages are published to "topics" or named logical channels. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe, and all subscribers to a topic will receive the same messages. The publisher is responsible for defining the classes of messages to which subscribers can subscribe. Topics are usually expressed in a URL-like notation. It is similar to creating a group. When subscriber subscribe to a topic it becomes a member of an event group. Topic is a pre-defined criterion and does not say anything about the specific content of an event. Example topic is "course2012".

In a content-based system, messages are only delivered to a subscriber if the attributes or content of those messages match constraints defined by the subscriber. The subscriber is responsible for classifying the messages. Topics are formed by
considering content of events, e.g., internal attributes of data structures. Application developers might use SQL, XPATH, and some other string API tools to handle complex topic strings. Example topic is "Course=DSMware and Grade=10".

Type-based system is an extension to content-based system. It ensures type safety at compile time. It does not filter according to a string topic but type as in the notion of programming languages.

Selecting different variant of publish/subscribe system depends on expressiveness and performance criteria. Content-based approach has higher expressiveness than topic-based approach and topic-based approach has higher performance than content-based approach. Application developers need to consider tradeoffs in scalability, expressiveness and quality of services depending on used architecture and implementation/protocols. In our implementation domain, which is GIS, we use topic-based approach. WFS are publishers and WMS are subscribers. We use NaradaBrokering as publish/subscribe system. The proposed system is explained in the following chapters.

There are some examples of publish subscribe systems. Among these are SIFT (Stanford Information Filtering Tool) [21], Microsoft’s Herald [22], SCRIBE [23] and IBM’s Gryphon [24]. Scribe is example of topic-based publish/subscribe systems. Gryphon is a content-based publish/subscribe system. Many applications such as stock quotes, network management systems, RSS feed monitoring, already benefit from this paradigm.

3.2 Systems Components to Create a Map

WMS and WFS are the fundamental services to create a still map images from geodata sets according to the open standards. Our earlier work on developing Web Service based WMS and WFS are presented in [25-27]. Geodata sets are served by WFS and rendered as map images at WMS. They have standard service interfaces and communication protocols. Here, we extend their communications with streaming data transfer capabilities, which is illustrated in Fig. 2.

Fig. 2 illustrates how the proposed architecture works to produce a map image from geographic features. The final outcome of the system (Fig. 2) is a map image. WMS clients request that image through WMS’s getMap service interface. Once WMS get this request, they create corresponding getFeature requests to WFS to get the feature data in GML format. After getting the data, WMS check and extract all the geometry elements such as points, line-strings, polygons etc., and converts them into appropriate image formats. WMS developers use any kind of graphics tools to create map images from those geometric features of data.

Developing WMS and WFS as web services enables them to be discoverable and used in third party distributed systems. However, efficient data transportation capability still remains as a challenge, because of the fact that web services are based on XML-based SOAP over HTTP protocol. In order to overcome such problem in the proposed distributed system framework requiring large scale XML-encoded geographic feature sets, we have investigated the possibilities of using topic-based publish-subscribe paradigms (which is mostly used in P2P systems) for exchanging data payload between web services. WFS using NaradaBrokering are called streaming WFS. When the NaradaBrokering is used, WFS are still queried with standard SOAP messages (requests) (arrow 1 in Fig. 2). However, the responses are published (i.e. streamed) to an NB topic as they become available (arrow 3 in Fig. 2). Arrow 2 shows the subscription stage. WFS send the “IP”, “topic” and “port number” to which the results will be streamed. The clients (WMS) subscribed to the same topic can receive the streams. WFS use MySQL in background and stream the results row by row (consider the relational tables) instead of waiting the calculation of the result set to be completed. The streaming enables I/O and CPU jobs to be overlapped and ends up with performance gains. The performance results are presented in our earlier work [26].

3.3 WMS as an Animation Server

We think that WMS are presently the most suited candidate for building a map server providing with the animated maps in accordance with the domain standards and web service standards. Although the standardization for animations is not mature yet [19], WMS’s specifications offer an appropriate standard for the sharing of data containing a temporal dimension. WMS provide same service interface called getMap for both animation and still map images, but with a little different parameter set for each.

WMS create static maps in pictorial formats from geo-data provided by WFS and store them in memory as an image array. Map animation is basically showing those images dated in a vertical frame one by one as a movie. If a WMS is capable of providing animation services for a layer (spatial data sets), then, some attributes are added under the layer definition in its capabilities file.
whose standards are defined as XML schema by OGC. If a layer (i.e. a map) is provided in time intervals this layer might then have a tag named "<Dimension>". According to the standards, multi-dimensional data objects are described with <Dimension> tag. Time is one of the dimension name defined in WMS capabilities file. If a time dimension is defined for a specific layer, that means geodata used to create that layer are spatiotemporal data and available in some time intervals and collected with a pre-defined periodicity. Examples of periodicities are once every second, once every year etc.

<Dimension> tag has some properties (name, units etc.) and a format as given below.

<Dimension name="time" units="ISO8601" default="2000-08-22"/>
</Dimension>

One layer might be available in multiple disjoint time intervals and those intervals might have different periodicities. At that time, WMS add additional lines to time dimension element as displayed below.

![Diagram of GetMap Request schema to be able to create streaming map movies](image)

The remainder of this section explains what a WMS does when it gets an animation (or map movie) request as given in Fig. 4.

In the values of time parameter given as examples above, first date defines the starting date and second date defines the end date of the available data collection.

The last value (ex. "P1D") defines the periodicity of data collection. According to last value in parameter time, WMS cut the time into multiple values and for each time interval it makes a request to WFS to get feature data in GML. See the successive requests (reqi) in Fig. 5, gml, represent corresponding responses from WFS for the requests reqi. img, are images created from corresponding gml.

User interaction with the system is achieved through browser based WMS clients. MapClient predefines the animation format in a particular style, define in what date ranges and in what time slices animation is needed. The appropriate query is created (e.g. Fig) and sent to WMS through its getMap web service interface. Once WMS get this query, it creates successive queries (req1, req2, ..., reqn) based on the time parameter in getMap request. Each of those queries is responded with gml (gml1, gml2, ..., gmln). For each gml a still map is created. Every still map corresponding to a time slice is stored in memory as a part of an image-array and displayed in a vertical sub-window simulating a camera film.
A Service Oriented Framework for Animating Big Spatiotemporal Datasets

be played depends on the time period and periodicity of the time intervals. To assemble the individual frames into an animation, there are different approaches according to application requirements. In the Internet world, widely used and well-known approach is using animated GIF89a files. The browser will be enough to open the animation, but the users do not have any control on the image such as stopping, pausing or play-backing the animation. The animated gif will play only in one direction. There is no way to make it play backward. PROG GMAP [28] is an example of gif animations. The Java Media Framework (JMF) [29] and Quicktime [30] provide with other approaches to assemble individual frames into an animation. JMF uses RTP sessions and need Java virtual machine installed on the client machines. That is used on Java based applications. OGC standards do not specifically and clearly define how to transfer and display animation. It only specifies the interfaces in terms of standard queries and output formats. In the proposed framework, we preferred to use IP multicast approach with JMF technologies.

Fig. 4. Sample GetMap request for WMS to create streaming map movies

3.4 Publishing Animated Map Images

After having the temporally related still map images created, we need to publish them as movie streams. WMS are access points for the distributed systems or any client to use map animation services. The first step in an animation is the creation of series of temporally related successive map image, which is explained in the previous chapter. The second step is playing these still map images as an animation, which is explained in this chapter.

WMS need successive map images for a period of time, in order to be able to create an animation. The time period and the periodicity of the movie frames are defined by parameter called "time" in GetMap request (Fig. 4). The number of frames to

Fig. 5. Detailed Streaming Map Movies architecture
An animation is produced as video streams. Map images are converted into sequence of video streams and published to an RTP session [31]. RTP sessions are formalized as <IPAddress, PortNumber> pairs. There are various video stream formats. The framework uses H.263 and H.264, which are well-known and widely used formats. Those animated video streams can be played and displayed on videoconferencing systems and collaborative environments such as AccessGrid (http://www.accessgrid.org/), but they are supposed to support H.263 and H.264 formats. The produced streams are published to multicast or unicast RTP sessions. Video streams can be delivered to a variety of platforms such as RealPlayer, Polycom and Access Grid [32]. The published video streams can also be displayed by any client building his own custom system and services to display the map video streams. The easiest way to display the movie map stream is connecting to RTP sessions by using a JMIF Client.

The quality of streams depends on some configurable parameters such as video format, frame rate and update rate. These parameters are set at the creation time, depending on the data and the application specific requirements.

4 Conclusion and Future Work

The work presented here has looked into the possibilities of extending the web based map services with time-series data as animated maps, and introduced a framework enabling integration of animation services to the distributed systems. At the core of the framework there is a WMS. It is actually an access point for the distributed systems to use map animation services. We have also extended the standard GIS web service communications with the topic based publish-subscribe communication approach. In the framework, GIS web services uses standard interfaces for the handshake, the actual data is transferred over the P2P overlay network provided by Naradabroking. After the handshake, communicating peers start transferring the data through the agreed upon broker (JP) and topic (any string). This approach enables us to create map images for partially returned data and get rid of the SOAP message creation overheads.

In the traditional tightly coupled client-server paradigm, the client cannot post messages to the server while the server process is not running, nor can the server receive messages unless the client is running. In case of transferring and rendering big spatial data through RPC messaging and web services, blocking IO and session timeout degrades the usability of web based applications. The proposed system gets rid of such communication latencies by using asynchronous communication.

The proposed framework is developed with open standards and Java technologies. Therefore, it can easily be enhanced and extended for application specific purposes, deployed on any platforms and integrated to the third party distributed system applications.

In the proposed system, movie streams (for map animations) are created on demand from the archived data sets. In the future we plan to enhance the system with the capability of archiving map animations. In that case, each archived map animation needs to be annotated with some parameters enabling them to be searched.

These parameters might be “temporal data layers from which movie streams are created”, “frame rates”, “starting-ending time/dates of the animation” and “periodicity of the data frames”.

References

Searching Frequent Itemsets by Clustering Data: Towards a Parallel Approach Using Mapreduce

Maria Malek and Hubert Kadima

EIST-LARIS Laboratory,
Ave du Parc, 95011 Cergy-Pontoise, France
{maria.malek, hubert.kadima}@eisti.fr

Abstract. We propose a new algorithm for searching frequent itemsets in large databases. The idea is to start searching from a set of representative examples instead of testing the 1-itemset, the k-itemset and so on. A clustering algorithm is firstly applied in order to cluster the transactions into k clusters. The set of the k representative examples will be used as the starting point for searching frequent itemsets. Each cluster is represented by the most representative example. We show some preliminary results and we then propose a parallel version of this algorithm based on the MapReduce Framework.

Keywords: Data Mining, large transaction bases, frequent itemsets, clustering algorithm, MapReduce.

1 Introduction

Association rules mining is a very known data mining techniques that aims to find relationships between items in large data bases that contain transactions. The problem of frequent itemsets has been introduced by Agrawal in 1993 and the well known Apriori algorithm has been proposed in 1994 [1]. This algorithm is based on the downward closure property: if an itemset is not frequent, any superset of it will not be frequent. The Apriori algorithm performs a breadth-first search in the search space by generating candidates of length K+1 from frequent k-itemsets. This algorithm is based on storing database in the memory. Unfortunately, when the database size is huge, both the memory use and the computational cost still be expensive.

In 2000, The FP-growth algorithm has been proposed [3], the idea is to use a frequent pattern tree data structure to achieve a condensed representation of the data transactions. This algorithm is based on a divide-and-conquer algorithm approach in order to decompose the problem into a set of smaller problems. The performance study shows that the FP-growth method is efficient and scalable for mining both long and short frequent patterns and faster than the Apriori algorithm.

In [5], authors discuss a depth first implementation of Apriori, the algorithm builds a tree in memory that contains all frequent itemsets: all sets that contained in at least minsup transactions from the original database, by adding one item