GeoTorrent: Optimizing GIS Web Services for Interactive Educational Use

Jeffrey A. Bergamini
Dr. Michael Haungs, Advisor
Department of Computer Science
California Polytechnic State University
San Luis Obispo, CA 93407
Phone (805) 756-2824 Fax (805) 756-2956
{jbergami,mhaungs}@calpoly.edu

Large online geospatial datasets for use with GIS are increasingly available and have many uses, including study in numerous academic and scientific fields. Access to this data in real time for pedagogical purposes would be immensely useful to educators. The academic lab environment presents unique requirements for data access: Its exploratory, multiplicative, and connective nature provides challenges and opportunities that have gone unexplored. Currently, these challenges make interactive educational use impractical. Also, student training requirements have been a noted problem. We present "GeoTorrent", a system that enables such use for K-20 educational institutions. GeoTorrent provides a client-side proxy server that manages OGC WMS connections in order to optimize user-perceived experience. GeoTorrent works alongside and transparently to WMS-compatible GIS client software, assembling a shared distributed P2P cache, intelligently prefetching data, and offering aggregated service information. Several novel approaches to caching and prefetching geospatial data make this possible. Preliminary experiments demonstrate substantial improvements in user response time and service availability, and a dramatically reduced impact on remote data sources. GeoTorrent also offers an increased ability for educators to share data with and among students, reducing the amount of student training required.

1. Introduction

Massive geospatial datasets are increasingly available on the Internet. A number of high-profile examples exist, e.g. NASA Jet Propulsion Laboratory's OnEarth [OnEarth] and the GLOBE Visualization Project [GLOBE]. Online distribution of these various datasets addresses a common problem: They are too large, expensive, and/or dynamic to duplicate and store locally, so they are maintained and provided online by the various agencies that collect and store the data. The Open Geospatial Consortium [OGC] has developed standards for the publication and access of geospatial data through web services, which are now nearly universally accepted.

Presently, there is one major consumer of these datasets that has not been specifically addressed. Massive scientific datasets would be of great educational benefit to educators and students at all levels. There are many possibilities for including GIS in education across a large number of fields and age groups, and these will only increase as more data become available. For example, imagine forestry, agriculture, or city planning students using these datasets to view and study any national forest, farmland, or city instantaneously from the classroom. The data is currently available. Unfortunately, the nature of educational use and access methods currently make such use impractical or impossible. The OGC standards do not specifically address the inherent challenges and opportunities of the educational environment.

We present "GeoTorrent", a system that enables the interactive use of data from OGC Web Map Service (WMS) servers by arbitrarily large groups of local users, through peer-to-peer caching, prefetching and service data aggregation, while conforming to and leveraging OGC standards. GeoTorrent was designed (and is described) with the K-20 educational lab environment in mind, but it may be useful in other situations as well. We introduce several novel approaches to the problem:

- **WHoMPyTS**, a conceptual model of WMS datasets that allows for flexible, client-neutral tile-based data caching and prefetching.
- **WMSmooth** is a heuristic for dynamically analyzing and modifying incoming WMS queries to make them suitable for caching and prefetching.
- GeoTorrent can run in multiple modes, from a completely decentralized **peer-to-peer** system, to a completely centralized **single server** system, and various **modes in between**, tailored to a given environment.
- GeoTorrent creates a **shared, distributed cache**, which may be redistributed on demand for **load balancing** and **resource allocation**.
- **WMS prefetching** reduces effective user response time by predicting and fetching future requests.
- In addition, GeoTorrent offers **aggregated service data** through a **web-based configuration console** which addresses the issue of student learning curve and training.

Preliminary experiments with GeoTorrent demonstrate substantial improvements in user response time and service availability, and a dramatically reduced impact on remote data sources. GeoTorrent also offers an increased ability for educators to share data with and among students, and reduces the amount of student training required.

In this paper, we give an overview of GeoTorrent and the problems it addresses. We then discuss some aspects of our implementation, evaluate its performance, review related work, and make suggestions for future work.

2. GeoTorrent

In this section, we give an overview of GeoTorrent and explain how its key features enable the use of GIS technology in K-20 educational settings.

2.1 Overview

We have designed GeoTorrent as a peer-to-peer, cooperative proxy caching and prefetching system for WMS data. GeoTorrent runs alongside, and is transparent to, WMS-compliant GIS client software in an educational lab setting. It exploits users' exploratory and multiplicative usage patterns, finding and using commonalities within these patterns to enable the practical interactive educational use of GIS. It also provides aggregated service data through a web-based console that facilitates data sharing and configuration for educators, reducing the amount of time required for student training.

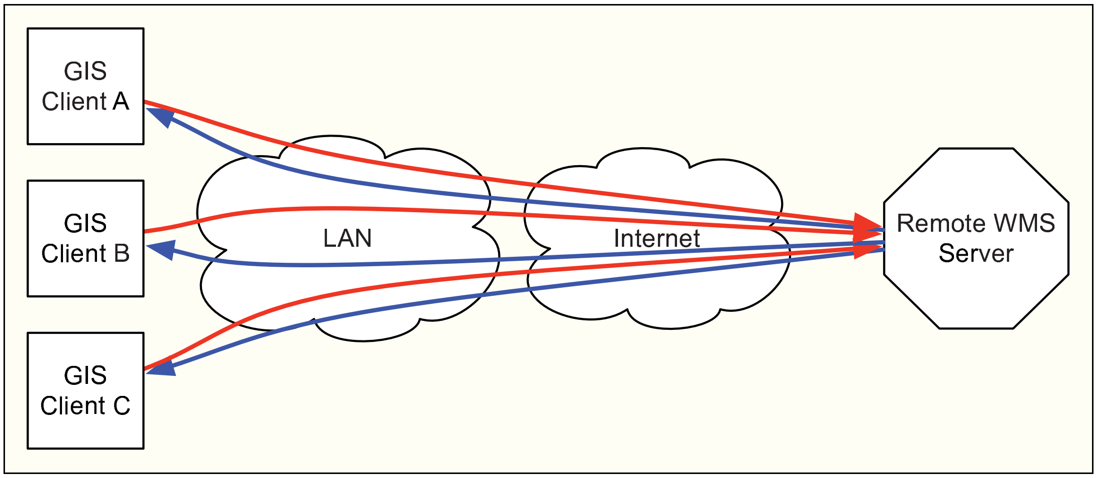

Figure 1: Traditional lab access to a remote WMS server

Figure 1 shows the traditional way in which GIS clients in an educational lab interact with remote WMS servers. Each client works independently of the others, resulting in the slow response times and denials of service we describe later. Figure 2 illustrates how GeoTorrent changes this situation, from the perspective of one client in a lab group.

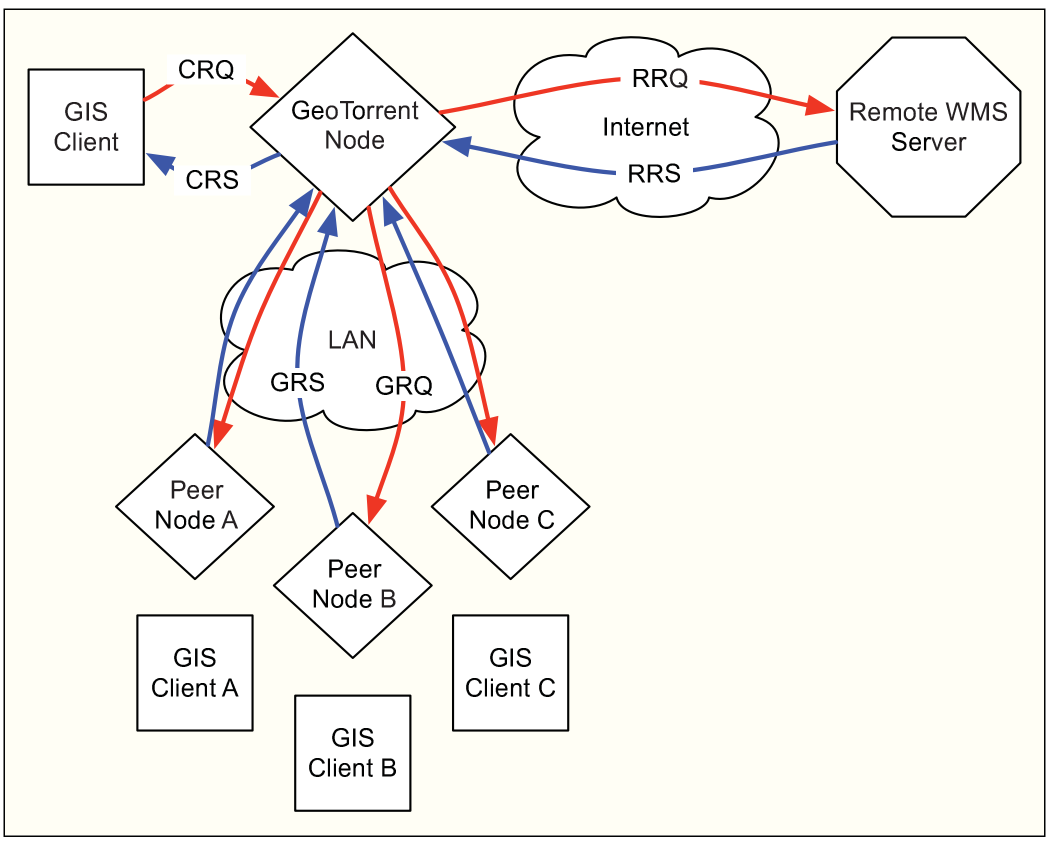

Figure 2: Access through GeoTorrent's P2P mode

A WMS client uses a GeoTorrent node as its exclusive WMS server. As a result, GeoTorrent can control how data is cached to be shared among the group, prefetched based on group activity, and offered through unified, aggregated service data. Each node in the GeoTorrent peer group acts as a proxy for one or more GIS clients, and a cache/prefetcher for the entire group. A client request (CRQ) made through GeoTorrent can be handled in three different ways:

1. If the client's GeoTorrent node has a local cache of the request, it is immediately returned (CRS).
2. If another node in the group has a cache, the client's node forwards the request (GRQ) to the node which has the cached response (GRS), and forwards this back to the client.
3. If no node has a cache, the requesting client's node forwards the request to actual remote WMS server (RRQ), returns the response (RRS) to the client, and caches it for the group.

Note several important aspects of using a remote WMS server through GeoTorrent:

- Clients make standard WMS requests.
- Each unique WMS request is made only once to the remote server, eliminating redundant load.
- GeoTorrent nodes need only service the rest of the lab group, unlike the remote server.
- The bulk of traffic is shifted to the LAN.

2.2 Current Challenges and Opportunities in Educational GIS

The requirements and environment for pedagogical use of geospatial data are quite different from those of scientific exploration. Current infrastructure and access methods fail to support interactive educational use. GeoTorrent addresses these challenges.

2.2.1 Limited Interactivity and Scalability

An interactive experience is vital for effective classroom GIS. Students often have limited lab time, or need to follow along with a lecture; they need quick, fluid responses. However, many publicly available servers exhibit excessive response times and unavailability when servicing the number of simultaneous requests that arise from even a single lab of students.

In section 4 we demonstrate several major problems that arise when lab workstations concurrently access public WMS servers: Response times rapidly decrease as concurrency increases, and some servers flatly deny service.

These results indicate that directly using remote WMS servers in an educational lab environment is impractical or impossible. The interactivity and scalability problem seems to arise from the multiplicative nature of the lab environment: Students are grouped together in labs, making requests to the same servers. This challenge naturally lends itself to a caching-based solution: Individual students are likely to request similar (if not identical) subsets of the data provided by remote sources. Direct access burdens servers with many redundant requests, since each student's client software communicates independently with the server, regardless of the fact that other clients likely have made or will make the same requests.

One of GeoTorrent's main functions is to build a *shared, distributed cache* of WMS data for use among the lab group, solving the concurrency and redundancy problems. In addition, the exploratory nature of educational use makes possible the idea of predictive data fetching. If students tend to be focused on a specific area, or are exploring in a certain direction, we can prefetch a useful amount of that data. Another of GeoTorrent's functions is to *analyze, predict and prefetch data* that is likely to be requested in the near future.

2.2.2 Lack of Infrastructure

Much of the literature regarding challenges in educational GIS focuses on the scarcity of resources [Bak05].

Schools may have underpowered workstations and/or limited connectivity. Combining a lightweight web-based client with GeoTorrent could ease the burden on individual student workstations and the school's Internet connection, allowing for the use of interactive GIS in schools with modest IT budgets.

Infrastructural limitations also prohibit centralized caching / prefetching systems. Schools may not be able to provide servers or system administrators locally. We propose taking advantage of pre-existing resources, viz. student workstations. GeoTorrent can run in a variety of configurations, from a fully distributed *peer-to-peer system* on student workstations, to a fully centralized *single server*, and *combinations in between*. P2P GeoTorrent exploits the multiplicative nature of the lab environment, turning it to an advantage rather than detriment.

2.2.2 Training

Another noted obstacle to classroom GIS is the student learning curve, which puts a drain on educators' time resources [Par01]. Educators need to minimize the amount of time and effort required to incorporate GIS into their classrooms. Including GIS in a lesson plan currently involves a significant amount of preparation: Client software must be configured to use the appropriate servers. Without an easily accessible service directory that students can use, teachers may spend too much setting up software, or teaching students how to do so.

In order to ease this burden, GeoTorrent includes a *service aggregation system*. Data source descriptions are combined into and accessible through a simple, dynamically-generated *web-based directory* that is available throughout the lab. Teachers can instantly make a data source available to all students. Students can utilize it with a simple copy-and-paste operation.

3. Design and Implementation

We have implemented GeoTorrent as a Java application that runs on student workstations (or on one or more servers) in a GIS lab. GeoTorrent acts as a local proxy WMS server. Peer-to-peer caching and prefetching are all performed transparent to clients. GeoTorrent nodes offer all the capabilities of the WMS servers for which they act as proxies, while working with peer nodes to build and prefetch a distributed cache of remote data as students browse.

Caching and prefetching WMS data, as well as doing so in a distributed environment, present unique challenges. We review some of the novel aspects of our design.

3.1 WHoMPyTS: Tile-Based WMS Caching and Prefetching

Normative WMS imposes no practical limitations on the image size and/or bounding box of a request. This presents a rather large problem for caching and prefetching: Clients may generate requests that are similar, but not identical, and are therefore difficult to share and prefetch effectively. To overcome this problem, GeoTorrent requires clients to follow a simple and flexible rule, which many clients already follow, for generating requests in a tiled fashion. In doing so, we create a new twist on the classic tiled pyramid storage system, which we have dubbed "WHoMPyTS". We describe WHoMPyTS in detail in [BH06].

GeoTorrent uses an extended WMS capability, as defined in [WMSspec], to notify clients that WHoMPyTS is available:

```
<VendorSpecificCapabilities>
   <GeoTorrent WHoMPyTS="true"/>
</VendorSpecificCapabilities>
```

Figure 3: Extended WMS Capability for WHoMPyTS

Note that GeoTorrent provides this simply as a notification. Clients need not be specifically aware of GeoTorrent in order to ensure that GeoTorrent will be able to fully cache and prefetch their requests. They need only follow the request formation rule on which it is based. A preliminary survey of tiling-based clients ([GoogleMaps], [TerraClient], [UdigTile]) indicate that most already do.

3.2 WMS Proxy Architecture and Query Smoothing

GeoTorrent acts as a "cascading" WMS server, one that retransmits the layers of other servers [WMSspec] – in other words, a proxy. GeoTorrent offers clients the WMS Capabilities documents of other servers, slightly modified so that clients make requests through GeoTorrent instead of the original servers. Clients see and use GeoTorrent as a normal WMS server, even though it is performing caching and prefetching optimizations.

3.2.1 Capabilities Documents

Adding a WMS server to GeoTorrent is as simple as pasting its `GetCapabilities` URL into GeoTorrent's configuration web page. GeoTorrent then fetches the server's Capabilities document and finds several important elements, including the title, `GetMap` URL, image formats, etc.

GeoTorrent uses these relevant bits of information to generate a GeoTorrent Capabilities document (GCD) for clients to use. When a server is added to GeoTorrent, it is assigned a unique ID. A WMS client fetches a GCD in the same manner as it would from a normal WMS server, through a link which is conveniently available for students on the configuration web page.

Generating a GCD is a relatively simple process of filling in a template XML document with information for a specific server. Upon receiving a `GetCapabilities` request, GeoTorrent fills in a copy of the template with the server's information and returns this dynamically-generated GCD to the client. Each layer element's `cascaded` attribute is incremented, per [WMSspec, pp. 23-24]. The `GetMap` URL is replaced in order to use GeoTorrent. In the case of OnEarth, its normal `GetMap` URL is:

`http://wms.jpl.nasa.gov/wms.cgi`

A GeoTorrent node running on host 10.10.10.3 at port 12345 alters this to

`http://10.10.10.3:12345/wms?remoteserver=JPLWorldMapService.OnEarth.jpl.nasa.gov`

when generating the GCD.

3.2.2 WMSmooth: A WMS Query Smoothing Heuristic

GeoTorrent's most important functions (caching and prefetching) are triggered by and dependent on receiving `GetMap` requests from clients. Each time a request is received, GeoTorrent checks the local and (if applicable) group caches to see if it can return a cached version, predict future requests, etc. The main requirement enabling both of these functions is the ability to match between client requests to cached or prefetched responses. In practice, this is not trivial, primarily due to floating point inaccuracy in the calculation of bounding boxes, both on the client and GeoTorrent. Bounding box elements of WMS requests often extend beyond the precision required by their respective geographic extents and image sizes, leaving insignificant differences which prohibit cache and prefetch hits.

Since GeoTorrent has no control over the way clients derive bounding boxes, it must have some way of finding “close enough” matches rather than an absolute ones. We propose a simple but elegant algorithmic solution: “WMSmooth”.

The idea behind WMSmooth is that GeoTorrent may discard insignificant portions of bounding box elements in client requests, for the purpose of cache lookup and prefetching. WMSmooth ensures that cache and prefetch hits occur when they should, regardless of insignificant differences in bounding boxes.

We discuss WMSmooth in further detail in [BH06].

3.3 Modes of Operation

GeoTorrent can run in a variety of modes, analogous to network topologies: Fully distributed P2P, fully centralized, and combinations in between. We discuss these modes, as well as their infrastructural suitability and requirements.

3.3.1 Peer-to-Peer

Recall Figure 2 for an illustration of P2P mode. A client communicates solely with its local GeoTorrent node, sending it WMS requests and receiving responses. Clients are unaware of any other GeoTorrent node, or of the remote server(s) with which its node communicates.

Group Communication: Multicast Messages

GeoTorrent group nodes communicate with each other through a combination of multicast and unicast (HTTP) messages. The details are beyond the scope of this paper, but we will review the concepts. There are several events that require a node to send a multicast message to the rest of the group:

Entry and Exit. Nodes announce their entry into a group, and those already present respond, to signal their presence.

Server Addition or Removal. When a user adds or deletes a server from GeoTorrent, the user's node notifies the rest of group, so that they can fetch the Capabilities document from the announcing node.

Cache Claim, Release and Migration. When a node caches a response from a remote server, it announces the cache to the group. When a node must relinquish a cached response due to lack of memory or cache migration, this is announced as well.

Fault Tolerance

GeoTorrent is able to detect and recover from unexpected loss of nodes. When a node finds another unreachable or does not respond, that peer is considered “exited”.

When a node is found to be missing, existing nodes invalidate the cached responses that it held. Since the failed peer's cache was equally as important as other nodes' caches, existing peers add the failed node's cache list to their prefetch queues, so that the missing peer's cache will be rebuilt among the still-existing nodes, if enough cache space is available.

Requirements and Suitability

P2P mode GeoTorrent can run in a lab environment with fairly modest hardware available. It requires only that the local network support multicasting and have an available multicast address and port through which the group may communicate.

3.3.2 Central Server

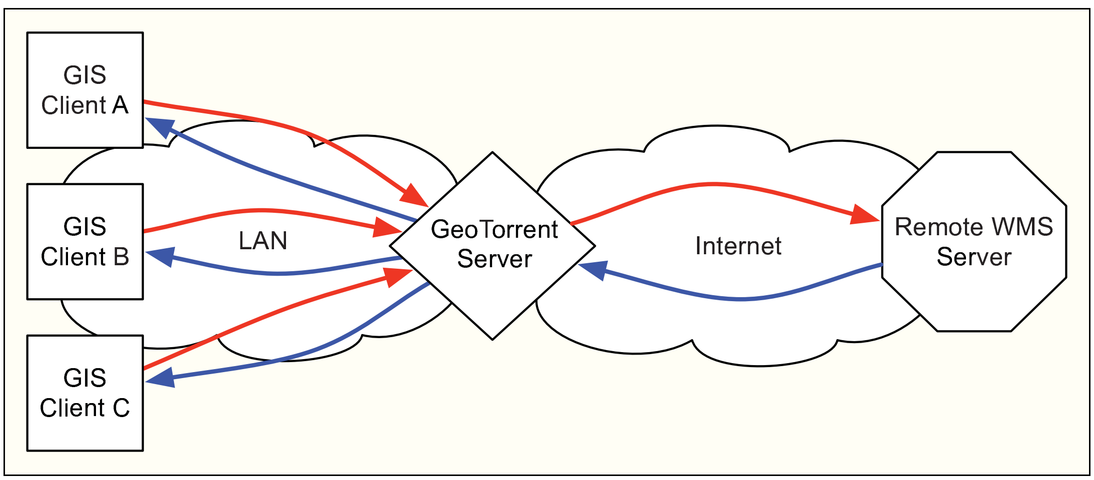

Figure 4: Central Server Mode

GeoTorrent may also run in a totally centralized, single server mode (Figure 4). Central server mode involves a single GeoTorrent node (now called a “server”) performing a subset of the tasks that nodes perform in P2P mode, for all clients instead of just one.

Requirements and Suitability

Central server mode is meant for a situation in which one dedicated server host machine is available. This server host should have significantly greater processing power and memory than the machines running the GIS client software.

3.3.3 Hybrid Modes

Besides the fully distributed P2P mode and the fully centralized central server mode, GeoTorrent may run in any number of hybrid modes between them. We describe two.

Dedicated Remote Server Nodes Mode

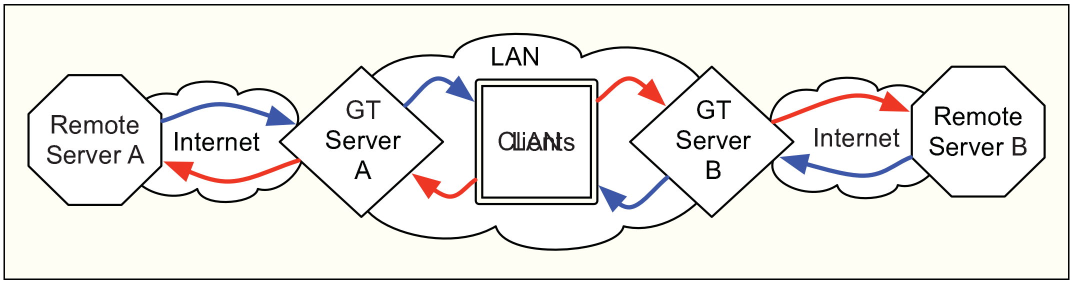

Figure 5: DRSN Mode

One possible hybrid mode is that in which a GeoTorrent server node is dedicated to servicing requests for only one remote server: Dedicated Remote Server Nodes (DRSN) mode. DRSN mode might be a good candidate for an environment in which multiple machines are available which have server-like processing and memory capabilities. It is a compromise between the possible bottlenecks of central server mode and the possible dependence on underpowered student workstations in P2P mode.

Partitioned P2P Mode

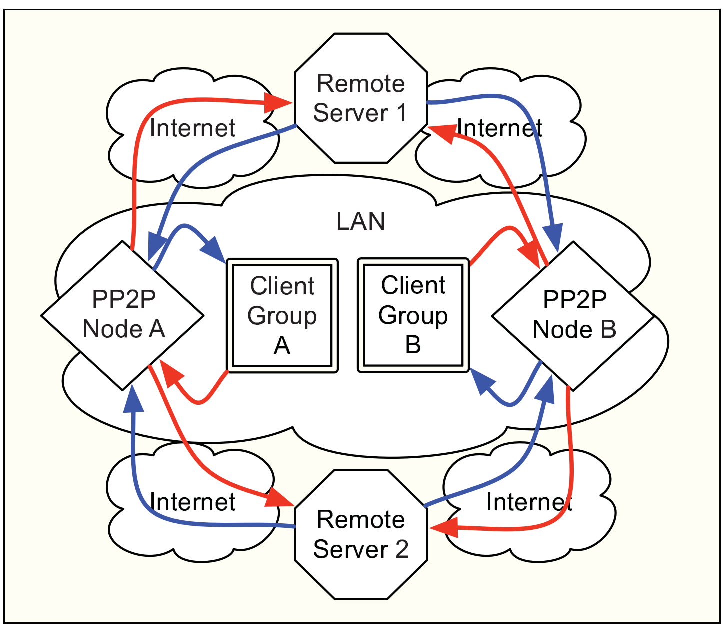

Figure 6: PP2P Mode

Another possible hybrid mode involves splitting up the P2P mode peer group while maintaining P2P communication: Partitioned P2P (PP2P) mode. The group of clients is divided into multiple groups, each of which communicates exclusively with one PP2P GeoTorrent node. The PP2P nodes communicate with all available remote servers, and with each other.

PP2P mode is compromise between DRSN and P2P modes, allowing each GeoTorrent node the opportunity to cache and prefetch for all remote servers. PP2P mode might be preferable to DRSN in cases where clients exhibit greater demand for a particular remote server, or if data from different remote servers require substantially different amounts of storage.

3.4 Shared, Distributed Cache

GeoTorrent nodes keep cached WMS responses. These can be returned to clients and other nodes very quickly. Nodes have user-set maximum cache sizes, and can migrate and/or intelligently replace cache entries as necessary.

3.4.1 Placement and Lookup

GeoTorrent places cached WMS responses in RAM, without any underlying database. Cache lookups are accomplished through a quick hashing function applied to WMSmooth-modified query strings.

RAM-store cache obviously is not persistent; when a node quits, its cache is lost. We think this is right in line with the WMS standard and with one of GeoTorrent's primary purposes, viz. providing data for interactive educational lab use. GeoTorrent's cached data need only persist (for example) throughout a lab session, or a period of days in which students work on a specific project. In addition, the WMS standard provides no ability to determine the temporal validity of data, so permanent caching could cause problems with data integrity (see Future Work).

3.4.2 LRU Replacement Policy

GeoTorrent uses a LRU (least recently used) cache replacement policy. If a node's cache is full and can't be migrated elsewhere, the node will trim the least recently used items from its cache until enough space is recovered.

Other research into caching geospatial data for interactive use has suggested that a locality-based cache replacement policy is more suitable [THLR01]. Locality-based caching places a priority on cache items close to the perceived user location. However, since in most cases each GeoTorrent node effectively serves multiple users, the idea of “user location” may not be meaningful. LRU allows for a more general, lab-wide view of cache use.

3.4.3 Migration / Redistribution

In the event that any node's cache is nearing its capacity level and other nodes have excess capacity, GeoTorrent performs a cache reallocation through migration. Note that this requires GeoTorrent to run in one of its P2P modes.

We use an 80/60/40 rule for cache redistribution: When a node's cache is more than 80% full, its least recently used cache entries are redistributed within the peer group until the node's cache utilization is down to 40%, if possible, or 60%, if necessary. The process occurs as follows:

- Nearly full node (N_0) announces cache redistribution, temporarily preventing any other node from doing so.

- Other nodes reply with two numbers: Their maximum cache sizes, and the unused portions thereof.
- N_0 runs through its least recently used cache entries for the next one (or possibly two) steps:
- N_0 sends any peers under 40% utilized an equal number redistribution requests. If all other nodes reach 40% before N_0 does, and N_0 has decreased to at least 60%, this ends the current migration process. If not, N_0 continues to the next step.
- If all other node caches were already at least 40% utilized or N_0 is still over 60%, N_0 performs round robin redistribution requests to the rest of the group under 60%, equally distributing requests among them until either they all reach 60% or N_0 reaches 40%.
- N_0 announces the end of this redistribution.

3.5 Prefetching

An important element of GeoTorrent's ability to decrease user response time is its prefetching mechanism. Nodes use the WHoMPyTS and WMSmooth guidelines, combined with statistics on incoming client requests, in order to predict and prefetch requests that are likely to come from clients in the near future. Each incoming WMS request is analyzed for prefetching, and if the analysis results in any predicted requests, these are added to a prefetch queue. Nodes process the prefetch queue with a lower priority than actual client requests, to avoid a negative impact on immediate user response time.

GeoTorrent offers two prefetching methods. The first and simplest is called the “surrounding squares” prefetching (SSP) method. SSP involves exactly what its name implies: As each request is received and passes the WHoMPyTS test, GeoTorrent quickly analyzes the request and calculates each of its surrounding tiles. Any of these which aren't already cached by the node/group, or scheduled for prefetching, are added to the front of the prefetch queue.

The other prefetching method is called “prevailing wind” prefetching (PWP). PWP is modeled after the idea of a prevailing wind measurement. Each GeoTorrent node may be processing multiple requests from multiple clients within a fairly short period of time. PWP attempts to determine a general direction in which a given client is browsing, and place a higher priority on prefetching tiles in that direction. PWP is still under development, so we refrain from discussing details.

Each GeoTorrent node contains a prefetch queue, which is populated as described above. The queue is processed by a thread running parallel to the WMS proxy. The prefetch thread iterates through the prefetch queue, sending out requests. Periodically, a node reviews its cache to find prefetched requests that have never been requested by a client. Any items older than a specified lifetime (set by user; default 30 minutes) will be removed. Given the tight locality of our prefetching methods, this is not likely to result in the loss of valuable prefetched data.

4. Evaluation

We are currently still improving GeoTorrent, including investigating the PWP prefetching algorithm. However, our initial experiments are very encouraging.

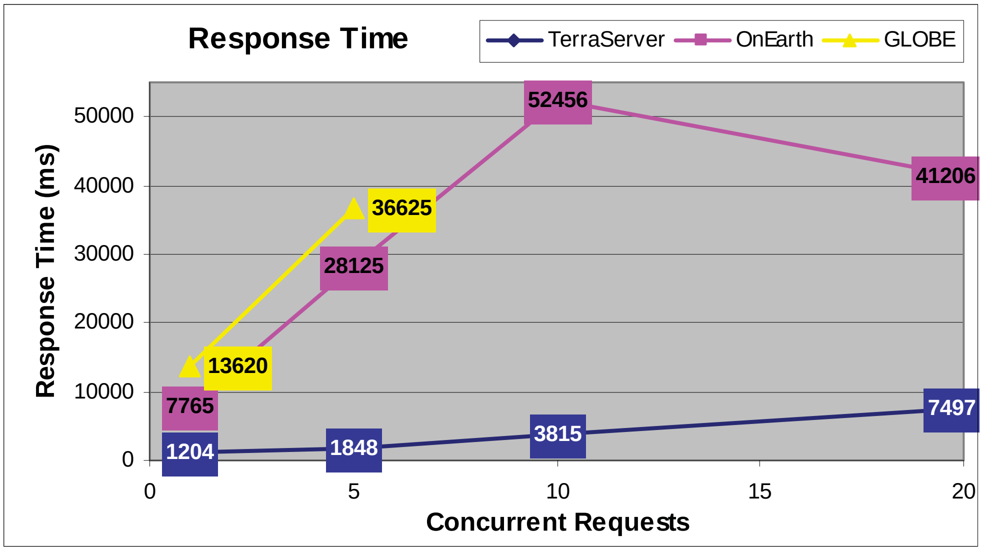

Figure 7: Traditional concurrent access results in unacceptable response times

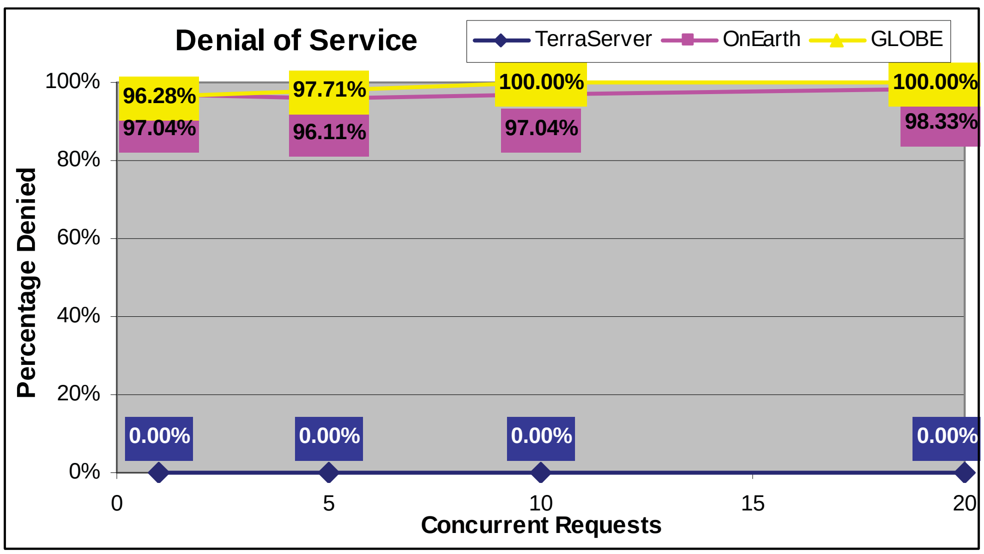

Figure 8: Denial of service prevents traditional use

Figures 7 and 8 demonstrate several major problems that arise when twenty lab workstations with unique IP addresses directly send requests to WMS servers with varying amounts of concurrency (i.e., traditional access). These problems are:

- **Response times rapidly decrease as concurrency increases.** OnEarth's response time degraded from almost 8 seconds per request to nearly a minute. GLOBE started out slow, then completely gave up and rejected all requests when fetching 5 or more concurrently. Note that it is common for tiling GIS clients to make 10-20 simultaneous requests in order to assemble a given view.
- **Denial of service.** TerraServer responded to every request we gave it, though response times increased by over 600%. OnEarth and GLOBE rejected over 90% of all requests, even

without concurrency. In a separate experiment, we determined that they began denying service when 6-10 workstations made requests simultaneously.

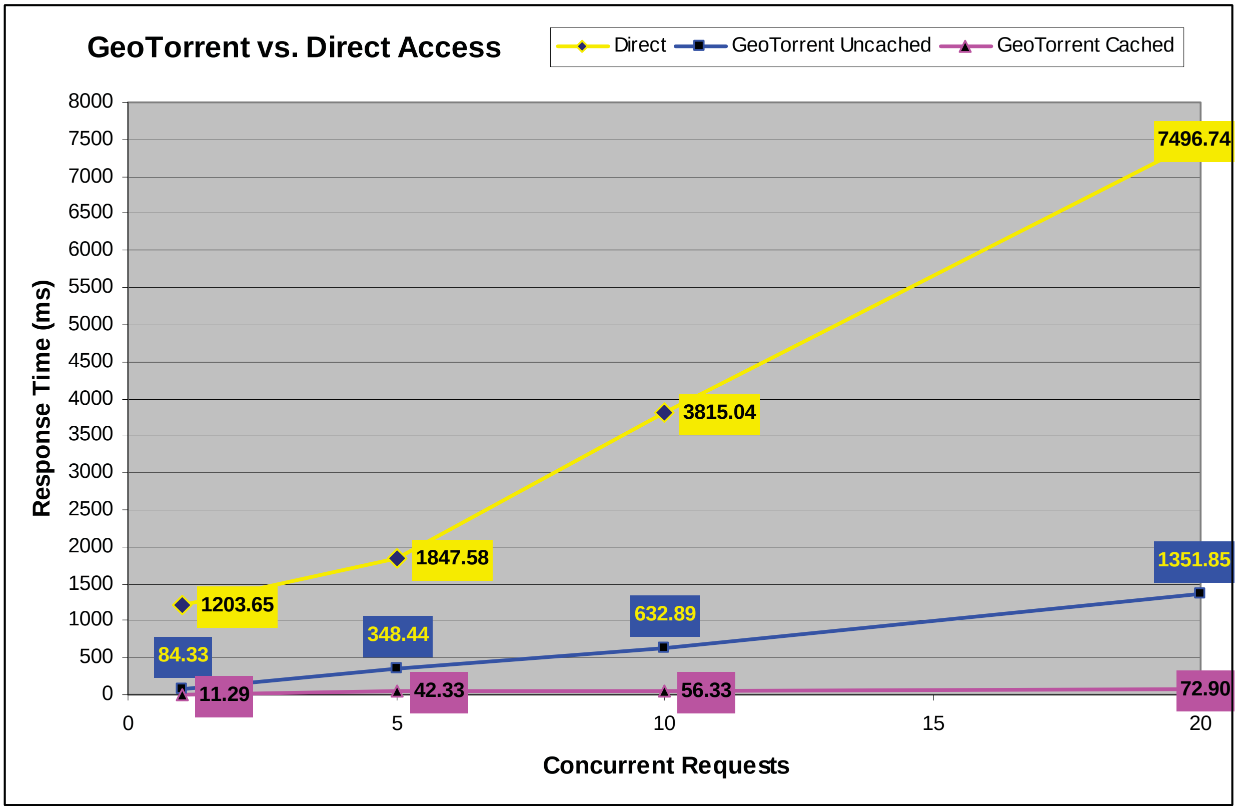

Figure 9: Results comparing GeoTorrent to traditional access

The next experiments compared traditional WMS use with GeoTorrent. We used the same twenty lab workstations, selecting randomly from the same list of WMS requests. We compared traditional ("Direct") access to two situations involving GeoTorrent: First, a situation in which GeoTorrent started with an empty cache ("GeoTorrent Uncached"). Then we tested the performance when GeoTorrent had already cached the requests ("GeoTorrent Cached"). Figure 9 demonstrates our results, using TerraServer as an example.

Dramatically Reduced Initial Response Time

GeoTorrent resulted in dramatic speed improvements, far exceeding the performance of what had originally proven to be the most resilient and responsive remote server. Even when starting with no cache, GeoTorrent achieved dramatically reduced response times. While processing 1 concurrent request per node, response times decreased by more than a factor of 14. Even at 20 concurrent requests, times were reduced by roughly 450%.

Scalable Distributed Cache

It is worth noting that the performance of the distributed cache had no problem scaling up to 20 concurrent requests per client. The response times from the cache kept pace with those of

TerraServer, increasing by 500-550% as we went from 1 to 20 concurrent requests. This sounds like a large increase, and in the case of direct access, it certainly made a noticeable difference (1.2 seconds vs. 7.5 seconds). However, GeoTorrent's cache response time grew from .01 seconds to .07 seconds. While similar in proportion, it is impressive that GeoTorrent was able to maintain response times of under 1/10 of a second.

Prefetching

In other, informal experiments, we have measured the performance of the surrounding squares prefetching model. While browsing with the Google Maps client, we tended to use just under half of the tiles that had been prefetched. This resulted in a visible improvement: In direct mode, every window scroll resulted in a waiting period while the client fetched new tiles. With GeoTorrent, those tiles usually appeared right away, since they had been prefetched and cached while we browsed.

5. Related Work

We review some of the more pertinent literature on proxy caching, prefetching / path prediction, and optimizing GIS for interactive use.

5.1 Distributed [Proxy] Caching

A fair amount of research exists on cooperative proxy caching of caching, including applications, limitations and suggestions. Most work on cooperative caching focuses on web proxies [Squid, IRD02, CBA04, DR02, LBG03]. We focus on several of the most applicable and interesting.

The Squirrel peer-to-peer web cache [IRD02] was designed to cache web pages among a group of machines by building a "scalable distributed hash-table" among the group. Web pages are associated with a given hash ID, and hash IDs are uniformly distributed among group members, for fault tolerance purposes. This results in each message passing through an average of three nodes. The goal of interactivity in GeoTorrent precludes routinely routing messages among multiple peers, and node failures are unlikely within the span of a lab period.

Other work attempts to avoid Squirrel's hash key assignment model. One such approach is modeled, oddly enough, after the hoarding mechanisms of squirrels (the mammals) [CBA04]. This work proposes a multi-agent system where peers work independently, but according to heuristics that result in a balanced resource allocation. They conclude that a "sniffing and burying" algorithm, where a peer "sniffs" several random possible locations before deciding to place a resource, results in a well-balanced allocation. We chose a reactive model instead, prioritizing immediate response over long-term balancing.

Some work suggests that cooperative proxy caching offers only negligible benefit for user response time, at least in web caches, due to the amount of cache misses [DR02]. However, our situation differs from standard web caching in that the peer group should repeatedly and reliably access the same or similar data.

General purpose web proxy caches, e.g. Squid [Squid], serve a purpose similar GeoTorrent's. However, due to the application-specific challenges we address with WHoMPyTS, WMSmooth, prefetching, etc. (as well as the educational environment's infrastructural limitations), normal proxies are not suitable.

5.2 Prefetching (Path Prediction)

The rise of mobile cell phones and hand-held computers in recent years has spurred interest in practical path prediction algorithms. However, they tend to focus more on regularity of movement based on daily or weekly patterns [LM95]. They also must take into account some things that don't necessarily apply to our situation, such as velocity of travel or road paths [KL03]. Others take into account other aspects of first-person actual human movement, e.g. frequented locations [AS03]. We needed a more generalized approach based solely on trends in recent access, perhaps better met by our prefetching models.

5.3 Other Work toward Interactive GIS

There has been a limited amount of work on producing a more interactive experience for GIS with remote datasets.

A 2001 paper entitled “A Systematic Approach to Reduction of User-Perceived Response Time for GIS Web Service” (SARUPRT) [THLR01] had a goal similar to ours. It proposed a caching and simple prefetching scheme to improve the experience of using GIS with remote data. However, there are fundamental differences between GeoTorrent and SARUPRT. SARUPRT focuses on “Internet users' slow communication speed” as the main problem, addressing a single-client, single-server model. It assumes a single user communicating with an infinitely capable server system, while we are concerned with performance loss due to many concurrent requests taxing several remote servers with finite resources. Additionally, SARUPRT focuses on a hierarchical data storage model for GIS data on a server. GeoTorrent has no control over the storage format of WMS servers. SARUPRT does not deal with OGC standards, and is tied to specific client and servers which are aware of the SARUPRT protocol and system.

Another paper detailed similar work (the SAND system) in 2002, entitled “Remote access to large spatial databases” [TBS02]. Like SARUPRT before it, SAND assumes a customized, non-OGC standard data access mechanism, tied to a specific client, server, and protocol. It introduces limited peer-to-peer elements, but in a centralized manner, not suited to the educational lab environment. SAND applies only to a situation in which clients are using a local dataset offered by a local server. SAND's peer-to-peer approach allows this local central data server to choose specific clients to be servers for specific bits of data. It does this by keeping a complete list of client data ownership, and publishing a directory service for the group of clients. This implies (1) a dedicated server machine, and (2) lots of directory requests made through that server. GeoTorrent doesn't impose any such constraint or require directory lookups upon each request. Finally, SAND's techniques are also limited to vector data. Images are not supported.

6. Future Work

The WMS standard doesn’t support content expiration. There is no way to tell whether cached or cascaded data is still valid. WMS could use something like HTTP’s “cache-control” headers, which define the temporal validity of data. This could be implemented as an additional element in each Layer element of a WMS Capabilities document.

If the database behind a WMS server already stores its data in a tiled fashion, WMS clients and servers could potentially take advantage of those tile boundaries for better performance, by making requests along those boundaries. This could also take the form of an additional element in the Capabilities document, advertising tile boundaries.

Many clients incorporate the idea of "zoom levels", where a user zooms in and out along discrete levels. If GeoTorrent could sense patterns among these levels, it could potentially prefetch data at nearby levels, anticipating and speeding up zooming operations.

7. Conclusion

We have introduced GeoTorrent, a system that enables interactive, educational use of GIS data from OGC WMS servers, through peer-to-peer caching, prefetching and service data aggregation. GeoTorrent addresses the current challenges and opportunities related to using GIS in an educational setting. Along the way, we proposed several novel approaches that enable WMS data to be cached and prefetched, in general and in a distributed system, all in a standards-compliant manner.

Preliminary experiments with GeoTorrent have demonstrated a substantial decrease in user response time and denial of service, and a dramatically reduced impact on remote data sources. GeoTorrent also offers an increased ability for educators to share data with and among students, as well as a decrease in the amount of student training required. These accomplishments provide a practical framework for incorporating GIS into education at many levels, which was previously difficult or impossible.

References

[BH06] Jeffrey A. Bergamini and Michael Haungs, *Enabling WMS caching and prefetching through implicit tiling and query smoothing* (working title). In submission to the 14th ACM International Symposium on Advances in Geographic Information Systems, November 2006.

[AS03] Daniel Ashbrook and Thad Starner, *Using GPS to learn significant locations and predict movement across multiple users*. Personal and Ubiquitous Computing 7 (October 2003), no. 5. (Springer-Verlag).

[Bak05] Thomas R. Baker, *Internet-Based GIS Mapping in Support of K-12 Education*. The Professional Geographer 57 (2005), no. i1, 44–47.

[BGS] Tom Barclay, Jim Gray, and Don Slutz, *Microsoft TerraServer: A Spatial Data Warehouse*. In proceedings of the 2000 ACM SIGMOD international conference on Management of Data 29, No. 2, May 2000. ACM Press

[BGS+02] Tom Barclay, Jim Gray, Eric Strand, Steve Ekblad, and Jeffrey Richter, *Terraservice.net: An introduction to web services.* Microsoft Research (Tech. Report), June 2002.

[CBA04] Sergio Camarlinga, Ken Barker, and John Anderson, *Multiagent systems for resource allocation in peer-to-peer systems*. Systems. In proceedings of the winter international symposium on Information and Communication Technologies WISICT 2004, Cancun Mexico, pp 173-178, January 2004.

[DR02] Sandra Dykes and Kay Robins, *Limitations and benefits of cooperative proxy caching*. IEEE Journal on Selected Areas in Communications, September 2002.

[GLOBE] GLOBE (Global Learning and Observations to Benefit the Environment) Visualization Program. http://www.globe.gov.

[GoogleMaps] Google Maps API. http://www.google.com/apis/maps.
[IRD02] Sitaram Iyer, Antony Rowstron, and Peter Druschel, *Squirrel: a decentralized peer-to-peer web cache*. In proceedings of the twenty-first ACM international symposium on Principles of Distributed Computing, July 2002.
[KL03] Hassan Karimi and Xiong Liu, *A predictive location model for location-based services*. In proceedings of the 11th ACM international symposium on Advances in geographic information systems, November 2003.
[LGB03] Prakash Linga, Indranil Gupta, and Ken Birman, *A churn-resistant peer-to-peer web caching system*. In proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security (2003).
[LM95] George Y. Liu and Gerald Jr. Maguire, *Efficient mobility management support for wireless data services*. Proceedings of the 45th IEEE Vehicular Technology Conference, Chicago, IL (1995).
[OGC] Open Geosptial Consortium, http://www.opengeospatial.org.
[OnEarth] JPL OnEarth, http://onearth.jpl.nasa.gov.
[Par01] Barbara Parmenter, *GIS in the classroom*. Learning & Leading with Technology 28 (2001), no. i7, 10.
[Squid] Squid web proxy cache, http://www.squid-cache.org.
[TBS02] Egemen Tanin, František Brabec, and Hanan Samet, *Remote access to large spatial databases*. In proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems (2002).
[Terra] Microsoft TerraService, http://www.terraservice.net.
[TerraClient] Microsoft TerraServer-USA, http://terraserver.microsoft.com.
[THLR01] Shengru Tu, Xiangfeng He, Xuefeng Li, and Jay J. Ratcliff, *A systematic approach to reduction of user-perceived response time for GIS web services*. In proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems (2001).
[UdigTile] Paul Ramsey, *1.1M7 -- WMS Tiling?* (post to udig-devel mailing list). http://www.mail-archive.com/udig-devel@lists.refractions.net/msg02460.html
[WMSspec] Open Geospatial Consortium Inc., *OpenGIS® Web Map Service (WMS) Implementation Specification (OGC 04-024) Version 1.3*. August 2004, http://www.opengeospatial.org/specs.