Feasibility of using existing web services for on-demand data access within distributed environmental decision support systems

Stuart F. Sheffield, Jonathan L. Goodall, Mohamed M. Morsy
and Alexander B. Chen

ABSTRACT

Web services providing machine-accessible interfaces to environmental data are now commonplace. Building on this, a current trend is to expand these web services to provide on-demand access to model and analysis services. This progression suggests the future possibility of cloud-based decision support systems (DSSs) integrating distributed data and analysis services delivered through a host of providers. Such distributed environmental DSSs have many potential benefits, but would require highly scalable and responsive web services. The objective of this study is to assess the current feasibility of building distributed environmental DSSs from existing web services in the United States. Results show that, of the many available web services providing information about soils, river network topology, watersheds, streamflow, etc., response times are often only a few seconds for a small project area, but can grow exponentially as the project area increases. On-demand watershed delineation remains a slow-to-respond service relative to the other services tested. Also, the results suggest the need to better co-locate servers near client applications to speed up response times. Collectively, these results provide specific areas where future research is needed in order to achieve the vision of on-demand distributed environmental DSSs.

Key words | decision support systems, environmental modeling, web services

Stuart F. Sheffield
Jonathan L. Goodall (corresponding author)
Mohamed M. Morsy
Alexander B. Chen
Department of Civil and Environmental Engineering,
University of Virginia,
Charlottesville,
Virginia
E-mail: *goodall@virginia.edu*

Mohamed M. Morsy
Irrigation and Hydraulics Department, Faculty of Engineering,
Cairo University,
P.O. Box 12211, Giza 12613,
Egypt

INTRODUCTION

The Internet has given researchers, scientists, and engineers the ability to quickly access and use data from different sources. The volume of data being produced in scientific fields is doubling yearly (Szalay & Gray 2006) and these data are increasingly being placed in online repositories and databases. Technologies like sensor networks and remote sensing have contributed to the surge of data with their increased use across scientific fields and generation of large, high-resolution datasets (Chen & Zhang 2014). With data availability growing, the need for tools that can automatically access information from different sources, then process and analyze the data to provide meaningful results for environmental management, has become essential (Hey & Trefethen 2005; Tarboton *et al.* 2014; Brodaric & Piasecki 2016; Chen & Han 2016).

Web services have become a popular method for automating access to scientific datasets. Web services are designed to communicate messages in a standardized format between computers over the Internet, allowing geographically separated computers to easily transfer data. Mineter *et al.* (2003) foresaw the need for the new generation of environmental applications to shift away from the desktop computer and toward more distributed resources interconnected through web services. Although web services

doi: 10.2166/hydro.2017.092

are being employed for data access, they are also progressively being used to produce derived data through more advanced analysis, visualization, and modeling performed on-demand based on user requests. Web service approaches have been proposed for various aspects of environmental applications including data analysis, visualization, and model simulation (Díaz *et al.* 2008; Granell *et al.* 2010; Booth *et al.* 2011; Feng *et al.* 2011; Goodall *et al.* 2011; Quiroga *et al.* 2013; Walker & Chapra 2014).

In recent years, several major federal agencies have begun to offer web services to access data stored on their servers. For example, the United States Geological Survey (USGS) has been making water data distribution and integration available via web services (Blodgett *et al.* 2016). Examples also include the Environmental Protection Agency's (EPA) Watershed Assessment, Tracking and Environmental Results System (WATERS) and the Storage and Retrieval and Water Quality Exchange (STORET), the USGS's National Water Information System (NWIS) and StreamStats application, and the United States Department of Agriculture's (USDA) Soils Data Access. The Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) project proposed web service standards to improve hydrologic time series access, as well as software for both server- and client-side data management within a distributed HIS (Goodall *et al.* 2008; Horsburgh *et al.* 2009, 2010; Ames *et al.* 2012; Tarboton *et al.* 2014). The standardization of web services makes automated access to heterogeneous data sources easier by providing a common interface for communicating between clients and servers.

The increased availability of standardized web services suggests that future decision support systems (DSSs) will be able to leverage common data, analysis, simulation, and visualization resources on-demand to support decision makers (Choi *et al.* 2005; Van Griensven *et al.* 2006; Buytaert *et al.* 2012; Harvey *et al.* 2012; Lu & Piasecki 2012; Laniak *et al.* 2013; Kumar *et al.* 2015; Galdiero *et al.* 2016). Although web services are commonly used now for data access within environmental DSSs as part of a preliminary, off-line, data gathering step, there are significant advantages to having a distributed system where web services are used to integrate data on-demand. One key advantage of the data or calculations being offered through a web service is that erroneous services can be changed and all clients will have access to the corrected information without the need to install new client-side software (Goodall *et al.* 2011; Buytaert *et al.* 2012). However, this ability raises a concern that reproducing past studies may be compromised due to unanticipated changes in underlying services. To address this concern, there would need to be clear and consistent ways to maintain versions of services and to alert users to updated services. Clever ways for archiving analyses, including the data and models that can be used to reproduce the analysis, will also be important. While admittedly more complex when dealing with distributed, service-oriented systems, these challenges of versioning, computational reproducibility, and provenance exist whether using a distributed or centralized DSS architecture.

Much of the prior research toward this vision has been directed at designing DSS web services themselves or evaluating architectures and their suitability for an environmental DSS (Matthies *et al.* 2007; Wagener *et al.* 2009; Sun 2013). However, due to the growing complexity of modeling real-world environmental problems, especially for the DSS, the rapid development of web services could play an even more significant role due to their ability to process the increasing demands of necessary datasets within a short time duration to support decision makers. Therefore, web service performance as it relates to providing these data will be an important benchmark as it relates to supporting a distributed environmental DSS. The objective of this research is to assess the feasibility of building a distributed environmental DSS using existing, authoritative, national-scale web services in the United States. This objective is explored through a stormwater management application used to identify data needs and map those data needs to available web services. A series of experiments were conducted to measure the response times of the service requests for different data access needs. The primary contribution of this research is to better understand the current state of web services for creating a distributed environmental DSS and to identify potential bottlenecks where future research and development could be directed in order to speed up web services and move toward the vision of distributed environmental DSSs.

The remainder of this paper is organized as follows. The background section introduces web services and a stormwater management application used to design a set of experiments for testing the feasibility of an environmental DSS. The methodology section describes the web services used for the analysis, as well as the specific experiments used to test the services. This is followed by results and discussion sections where results from the individual experiments are presented and discussed in terms of performance, reliability, and variability. Finally, conclusions from the study findings are presented, along with suggestions for future research building from this study.

BACKGROUND

Web services

Web services are defined as software systems designed to support machine-to-machine interaction over a network. There are two roles defined in a web service: a service provider and a service consumer or client. The service provider creates the web service, publishes access information, and registers what is available to the client. The client must find and invoke the web services to access available information or features. Two common methods to implement web services are Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). SOAP is a protocol or standard for exchanging structured information while REST is an architectural style. Applications that employ REST principles are called RESTful and were used for the web services tested in this study. To be called RESTful, applications must satisfy six constraints defined by Fielding (2000). These constraints include a separation of client and server, a lack of client storage on the server between requests, and a uniform interface.

RESTful services can be accessed by the client much the same way that internet browsers load web pages. Resources, such as information and data, are requested through a Uniform Resource Identifier which can be contained in a Uniform Resource Locator (URL). Additional options such as response formats and search parameters would also be contained in the URL. A general workflow for web services involves the client first invoking the web service by sending a message or request to the provider over the network using a URL with all necessary identification information. The provider reads the message, obtains the requested information, and sends a message back to the client containing the requested information over the network.

Stormwater management application

Overview

A stormwater management application is used to examine the feasibility of existing web services for data access within environmental distributed DSS. The application in this study, for Virginia, is a prototype, which could potentially be applied to other study areas in the USA.

In Virginia, Virginia Stormwater Management Program (VSMP) regulations, like other state regulations, require construction projects to account for stormwater runoff impacts from increased impervious surfaces in order to prevent erosion, flooding, and water quality degradation. Organizations have traditionally constructed onsite stormwater Best Management Practice (BMP) structures, such as detention ponds and/or bioretention facilities, to mitigate these stormwater impacts. Recently, changes to the regulations allow for nutrient credit purchases as an alternative to onsite BMP construction. This new option allows pollutant dischargers to purchase credits from off-site sources to offset what would be treated onsite.

To qualify for the use of nutrient credits, a project must meet at least one of the following criteria determined by the Code of Virginia § 62.1-44.15-35. First, the project area must contain less than five acres of disturbed land. Second, the post-construction phosphorus control requirement must be less than 4.54 kg per year. Third, if the first two criteria are not met and if the applicant can demonstrate onsite control of at least 75%, the remaining required reductions can be met through the purchase of nutrient credits. If the project discharges into a local watershed with an established nutrient total maximum daily load (TMDL), nutrient credits may still be purchased provided that the use of the credits do not prevent compliance with the local limitation.

There are several calculations needed to check whether or not a project meets the eligibility criteria. A key calculation is the determination of the amount of phosphorus generated by the site. Traditionally, this is done using the procedure described in the Virginia Runoff Reduction

Method (VRRM) using general site information, such as total acreage, soil types, land cover, and BMP types, and a regression equation to estimate phosphorus runoff. An alternative method is to use a surface water model, such as TR-20, to estimate runoff and pollutant loads. The model requires the drainage area, SCS curve numbers, and concentration times. These watershed properties can be estimated from land use, soil, and elevation data for the watershed.

This stormwater management application offers a typical example of data needs within environmental DSS. While each application will have unique needs, many will require soil, land use, watershed, and stream properties like the example application described in this paper. The work of examining the feasibility of using existing web services for on-demand data access is essential for building on-demand distributed environmental DSS in the future. Despite this being a fairly simple analysis, it still requires a broad set of input data from a variety of data providers, as described in the following section.

Workflow and data description

The workflow and summary of data needed to enable the example application are provided in Figure 1 and Table 1. The first step is to gather site information. It is assumed that the user will provide some of the inputs to the application, including the coordinates of polygon vertices and the project boundary, while the hydrologic unit code (HUC) and nutrient TMDL (total maximum daily load) of the site could be obtained from EPA WATERS. Other location information (i.e. project area, disturbed acreage, latitude, and longitude) is to be provided by the user. The second step is to build the pollutant runoff model. The required data for this step includes annual rainfall for the project site along with land cover, soil, and watershed information. These data could be gathered from EPA WATERS, USGS StreamStats and USDA Soils Data Access. The third step is for the user to provide the pricing and bank locations for nutrient credit purchasing. These data, combined with the outputs from the model in step 2, provide the decision-makers with actionable information that can be used to decide whether to purchase nutrient credits or build onsite structural BMPs. The following section describes these web services and how they can be used for data access to support the stormwater management application.

Service descriptions

Three service providers are available to provide many, but not all, of the data required to build a DSS for the example application (Table 1): EPA WATERS, USDA Soils Data Access, and USGS StreamStats. Each of the services are maintained by a federal agency and are open for public access.

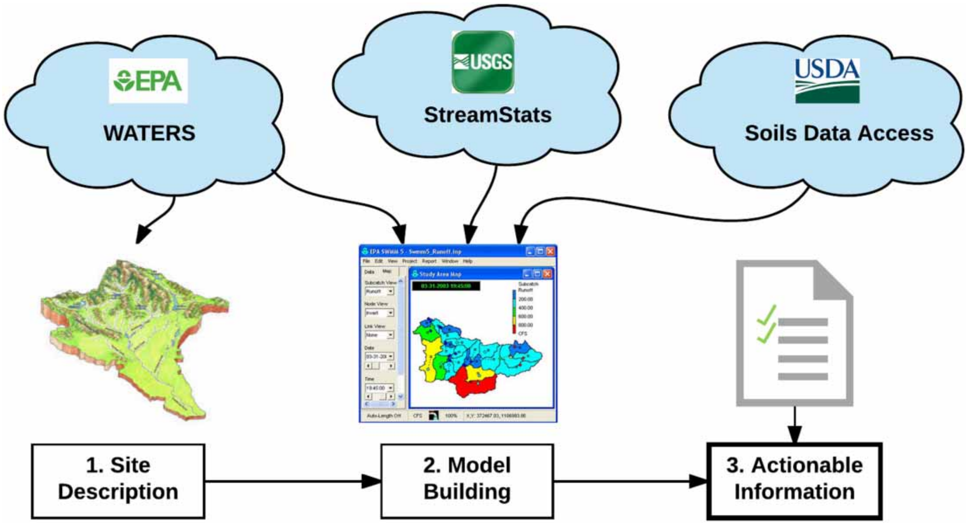


Figure 1 | Workflow of the stormwater management DSS application.

Table 1 | Information needed to support the stormwater management DSS application

Steps	Required information	Data needed	Source	Web service
1. Site description	Location information	Project area	User input	n/a
		Disturbed area	User input	n/a
		Location (city/county)	User input	n/a
		Latitude and longitude	User input	n/a
		4th order HUC	NHD	WATERS
		Nutrient TMDL	303(d) List	WATERS
2. Model building	VRRM data inputs	Annual rainfall	Watershed characteristics	StreamStats
		Land cover	NLCD	No
		Soils information	SSURGO	Soils data access
	TR-20 data inputs	Drainage area	Watershed boundary	StreamStats
		Channel length	NHD	WATERS
		Rainfall amount	Watershed characteristics	StreamStats
		Soils information	SSURGO	Soils data access
		Land cover	NLCD	No
3. Actionable information	Pricing		User input	n/a
	Bank locations		User input	n/a

EPA WATERS

EPA WATERS provides water quality information from various EPA sponsored programs and links it to the national surface water network (Environmental Protection Agency 2015). The surface water network is based on the National Hydrography Dataset Plus (NHDPlus). Users can locate dischargers, view water quality monitoring results, impaired water reports, and perform general stream navigation. The web services made available by WATERS also expose components used to perform complex analyses on the supporting datasets, such as NHD, NHDPlus, and the Watershed Boundary Dataset (WBD). Included in these services are the point indexing service and the upstream/downstream search service, among several others. The point indexing service links a coordinate location (expressed as a latitude and longitude) to the NHDPlus flow line network. The upstream/downstream search service provides the ability to navigate upstream or downstream of a user provided distance from a point on the network and returns a list with any events encountered during the traversal.

USGS StreamStats

StreamStats was developed by the USGS to provide users with several analytical tools that are useful for water resource planning, engineering, and design purposes. The web services provided through StreamStats can be accessed using the StreamStats service browser interface or simply through a web browser. The URL for a service request includes the service name, inputs required by the service, optional response formats, and which measurement variables the client wants to include in the output. These services allow the client to obtain the basin boundary, characteristics, and streamflow for a selected location on a stream network.

StreamStats is built partly on ArcHydro, a data model and tools for hydrologic data processing within a Geographic Information System (GIS) (Maidment 2002). Access through the network is provided through ArcServer. Elevation data is derived from the National Elevation Dataset (NED). This dataset was adjusted so that the stream channels correspond to those represented in the high-resolution version of the NHD, and so that watersheds correspond to those delineated in the WBD. After basin characteristics are measured, values are input into the National Streamflow Statistics Program, which is a program that uses USGS regression equations to estimate streamflow statistics for points along a river network.

USDA Soil Data Access

USDA's Soil Data Access web services were developed in order to meet objectives that were not being met by the

Web Soil Survey and the Geospatial Data Gateway. One of its objectives was to provide a way to request the data for an area of interest of any size in real-time. Currently, the Soil Data Access services return spatial and tabular data using separate requests. The spatial data request requires the user to supply an area of interest. The tabular data request takes as input a set of map unit keys and returns the desired tabular data for those map units.

Soils Data Access offers several options for accessing the spatial and tabular data. Users can access tabular data via SOAP or REST/POST requests. The tabular service includes a RunQuery method that returns XML data for one or more SQL statements. Users can request spatial data in different coordinate systems including WGS84, NAD83, UTM (Universal Transverse Mercator), and Web Mercator. The tabular service includes a RunQuery method that returns XML (eXtensible Markup Language) data for one or more SQL statements. The spatial services follow the Open Geospatial Consortium (OGC) Web Feature Service (WFS) standard and include GetCapabilities, DescribeFeatureType, and GetFeature. For this study, the GetFeature method using the WGS84 coordinate system was used. Two layers accessible through the GetFeature method are the mapunitpoly and mapunitpolyextended layers. The mapunitpoly layer contains identifying information about soil map units and the mapunitpolyextended layer contains more specific information like Hydrologic Soil Group, moisture, and slope.

METHODS

Five experiments were designed to test response times for essential data access queries required in the stormwater management application described earlier. Key parameters for each query were varied to measure their impact on response time. Experiment 1 was designed to test how the USDA soils web services responded to increasing study area sizes. Experiment 2 was designed to test how the distance between the project location and the river affected response times for the point indexing service from WATERS. Experiment 3 was designed to test how the downstream search distance affected response times for WATERS' upstream/downstream search service. Experiment 4 was designed to test how the size of the watershed affected response times for StreamStats services. Finally, Experiment 5 was designed to test how the client's location affected response times for all of the services. Details of the steps taken within each experiment are described in the following subsections.

All experiments were run using virtual machines (VMs) provided through Amazon Web Service's (AWS) Elastic Compute Cloud (EC2), specifically the t2.micro instances and the Ubuntu Amazon Machine Image (Table 2). The t2. micro instance at the time of this experiment featured high frequency Intel Xeon processors, and had a burstable performance that constantly provides a baseline CPU (central processing unit) performance but has the ability to burst above the baseline when required. There are no bandwidth limits for the t2.micro instances. The specifications of the t2.micro instances are shown in Table 2. AWS allows VMs to be created in one of three AWS server hosting locations in the USA: Northern Virginia, Oregon, or Northern California. Unless specified, all experiments were conducted using VMs located in Northern Virginia. Once the VM was running, Python scripts for each experiment were moved to the VM's local directory and run from the command line. The *urllib2* library was used to request URLs within the Python scripts. Timers, from the *time* library, were set before and after the URL request was made and returned. The service response time was defined as the difference between when the URL request was made and when the response was returned. Results were output to a comma-separated file that was saved on the VM. Each experiment was run multiple times to assess the variability in service requests. After copying the result files to a local computer, Python's *Pandas* library was used to analyze the data and the *matplotlib* library was used to visualize the data. Bar charts with error bars were made for each experiment to show the mean and one standard deviation around the mean for the response times.

Table 2 | AWS EC2 t2.micro specification

Model	vCPU	CPU credits/hour	Memory (GiB)	Features
t2.micro	1	6	1	High Frequency Intel Xeon Processors, Burstable CPU

CPU Credit: One CPU credit is equal to one vCPU running at 100% utilization for 1 minute.

One-way Analysis of Variance (ANOVA) was used to test the null hypothesis that population means for the response time between several groups were equivalent. If this null hypothesis were accepted, then all groups are considered statistically similar. However, if the null hypothesis were rejected, then at least one of the groups is significantly different from the others. ANOVA tests do not indicate which group is different; therefore, a post hoc test was required. The Tukey's Honest Significant Difference (HSD) test was used to identify groups whose differences exceed the expected standard error, indicating which group is significantly different. Both of these statistical analyses are available through the R software package, which is widely used for statistical computing and graphics (R Development Core Team 2008). The ANOVA tests and the post hoc testing using the Tukey's HSD method were completed using R version 3.3.1 and an alpha level of 0.05.

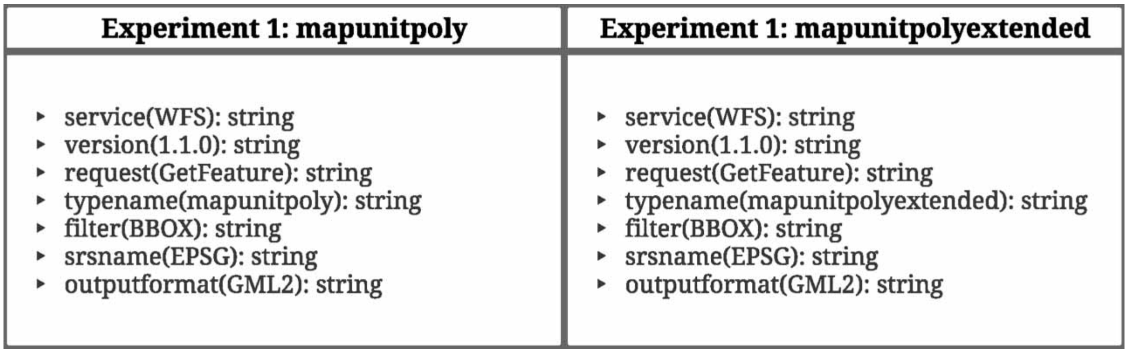
Experiment 1: soils data access

In order to conduct Experiment 1, two URL requests were made: one for the mapunitpoly service and one for the mapunitpolyextended service. The parameters specified in both requests are summarized in Figure 2. Both URLs request the GetFeature method, output in GML2 format, projected into WGS84 coordinates, and the default service type and version for the Soils Data Access Service (Figure 2). The GetFeature method returns a feature collection for a layer for an area of interest. A bounding box defines the area of interest in this experiment. The coordinates for the bounding box were varied to create polygons of approximately 1, 10, 100, 1,000, and 10,000 acres to test the response times as the study area increases. As shown in Figure 3, the mapunitpoly service was requested first, followed by the mapunitpolyextended service after the mapunitpoly response was returned. The difference between these two services is in the number of attributes returned for each feature. The mapunitpoly service returns seven attributes per feature while the mapunitpolyextended service returns 44 attributes per feature. Each URL was called 25 times in this experiment to measure the variability in response times.

Experiment 2: distance from stream network

Two URL requests were made in Experiment 2: one for the point indexing service and one for the upstream/downstream search service. The point indexing method was set to RAINDROP mode in order to force downhill travel to the river network. The point geometry represents the starting location that could be a project's most downhill point. The maximum indexing distance was set to the default value of 2 km as shown in Figure 4. The OutputPathFlag was set to FALSE. The coordinates for the point geometry represent distances 0.5, 1, 2, 3, and 4 km away from the nearest, downhill flowpath. The upstream/downstream search service was given a start COMID value, a unique identifier for a feature within the National Hydrography Dataset (NHD), which was obtained from the output of the point indexing service.

The stop distance was set to a constant 25 km from the initial point indexing location for all trials. The traversal summary was set to be downstream, mainstream (DM). The traversal summary, flowline summary, and 303(d) event summary lists were returned for all trials. The URL for point indexing was built and requested first. Upon return of the



Experiment 1: mapunitpoly	Experiment 1: mapunitpolyextended
▸ service(WFS): string ▸ version(1.1.0): string ▸ request(GetFeature): string ▸ typename(mapunitpoly): string ▸ filter(BBOX): string ▸ srsname(EPSG): string ▸ outputformat(GML2): string	▸ service(WFS): string ▸ version(1.1.0): string ▸ request(GetFeature): string ▸ typename(mapunitpolyextended): string ▸ filter(BBOX): string ▸ srsname(EPSG): string ▸ outputformat(GML2): string

Figure 2 | URL parameters for the web services tested in Experiment 1.

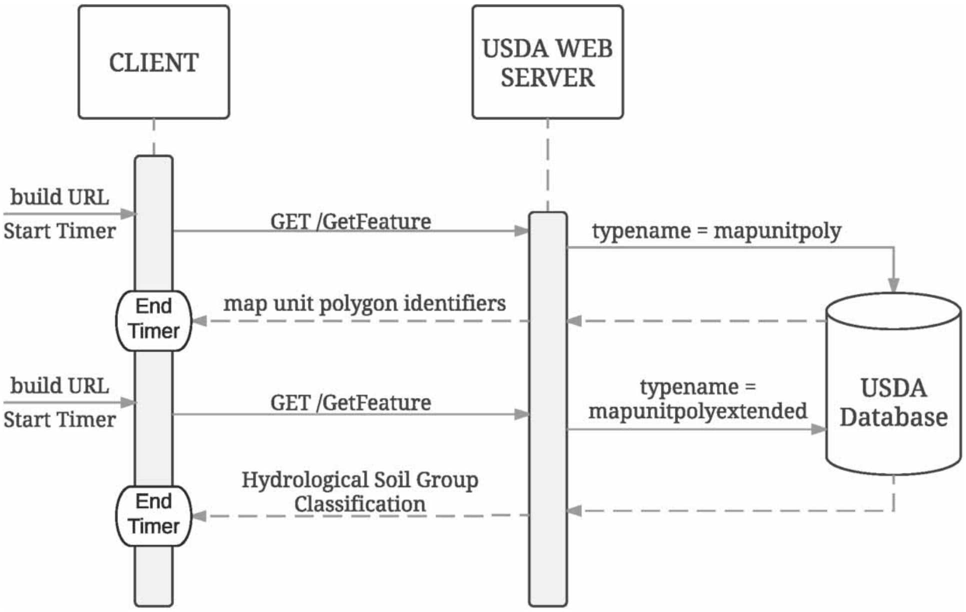
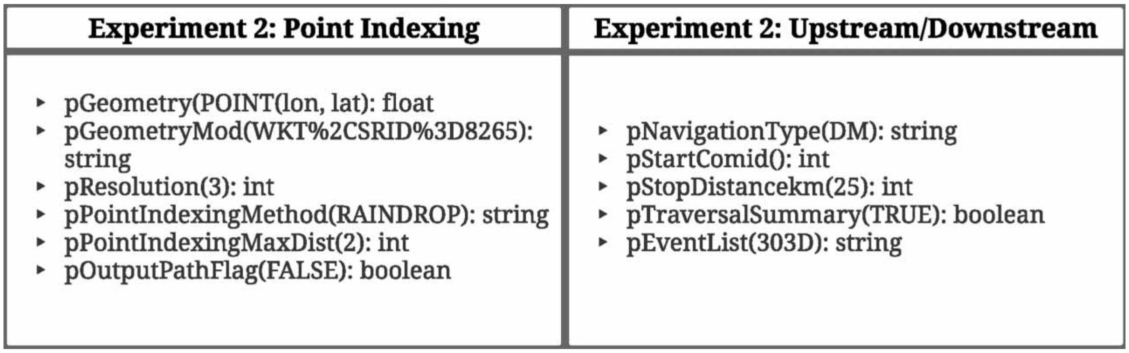


Figure 3 | Sequence diagram for web services tested in Experiment 1.



Experiment 2: Point Indexing	Experiment 2: Upstream/Downstream
▸ pGeometry(POINT(lon, lat): float ▸ pGeometryMod(WKT%2CSRID%3D8265): string ▸ pResolution(3): int ▸ pPointIndexingMethod(RAINDROP): string ▸ pPointIndexingMaxDist(2): int ▸ pOutputPathFlag(FALSE): boolean	▸ pNavigationType(DM): string ▸ pStartComid(): int ▸ pStopDistancekm(25): int ▸ pTraversalSummary(TRUE): boolean ▸ pEventList(303D): string

Figure 4 | URL parameters for the web services tested in Experiment 2.

data for the point indexing service, the URL for the upstream/downstream search service was built and requested (Figure 5). Both URLs were requested 25 times for this experiment to measure the variability in response time.

Experiment 3: stream network search distance

Two URL requests were made in this experiment: one for the point indexing service and one for the upstream/downstream search service. The URL parameters are summarized in Figure 6. Downstream search distances of 1, 5, 10, 25, and 50 km were used for the trials. All other variables used in Experiment 2 were held constant in Experiment 3. In contrast to Experiment 2, the input point geometry was set to a single location that was constant for all trials. The service call and timing sequence remained the same as Experiment 2 and are detailed in Figure 5. Each URL was requested 25 times for this experiment due to the longer response times of this experiment.

Experiment 4: watershed properties

Two URL requests were made in Experiment 4: one for watershed delineation and the other for basin

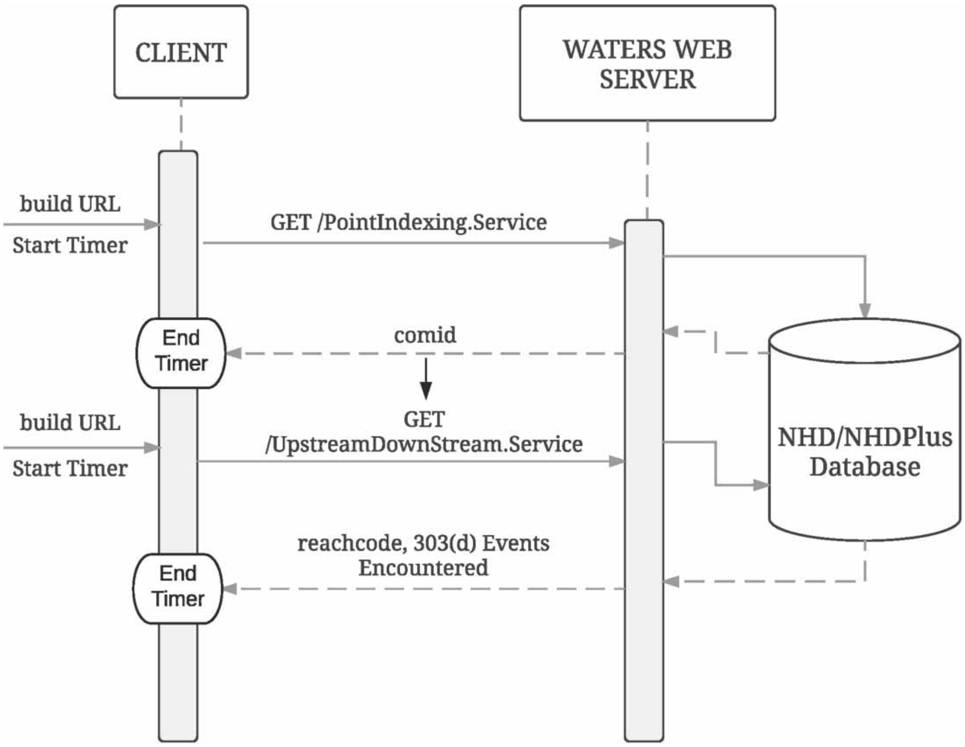
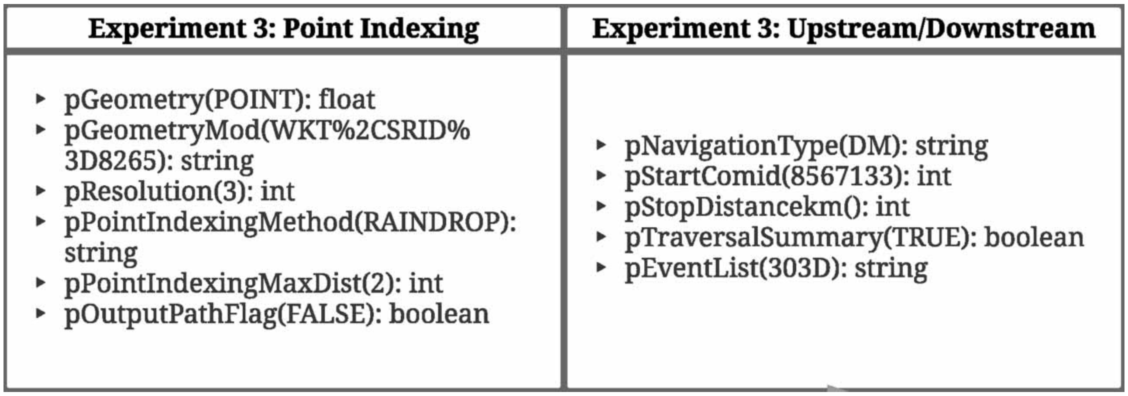


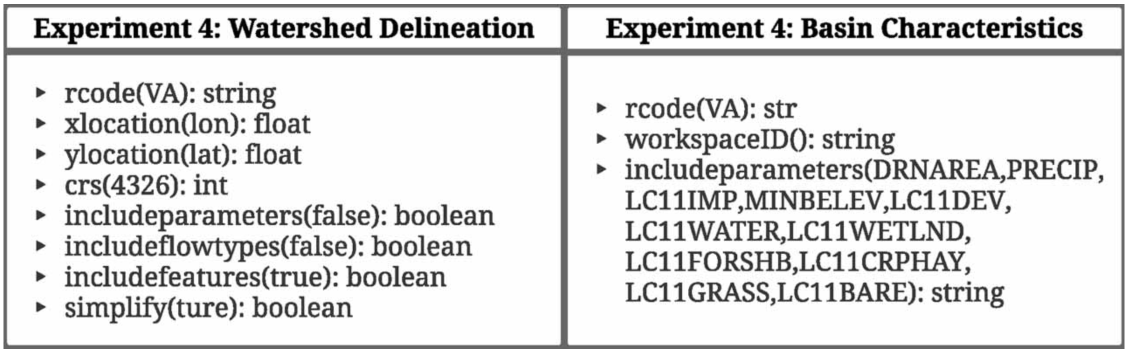
Figure 5 | Sequence diagram for the web services tested in Experiments 2 and 3.



Experiment 3: Point Indexing	Experiment 3: Upstream/Downstream
▸ pGeometry(POINT): float ▸ pGeometryMod(WKT%2CSRID%3D8265): string ▸ pResolution(3): int ▸ pPointIndexingMethod(RAINDROP): string ▸ pPointIndexingMaxDist(2): int ▸ pOutputPathFlag(FALSE): boolean	▸ pNavigationType(DM): string ▸ pStartComid(8567133): int ▸ pStopDistancekm(): int ▸ pTraversalSummary(TRUE): boolean ▸ pEventList(303D): string

Figure 6 | URL parameters for the web services tested in Experiment 3.

characteristics of the watershed. The URL parameters are summarized in Figure 7. The coordinates for x-location and y-location represent the longitude and latitude of a point on the stream network. For this experiment, the coordinates were varied to produce watersheds of 200, 800, 2,500, 25,000, and 110,000 acres. Both services offer options to include different lists in the output. The URLs shown in Figure 7 detail the selected lists. Upon completion, the watershed delineation service returned a workspaceID. The workspaceID was used in the basin characteristics service to return specific watershed information. For this experiment, only a select number of basin characteristics were returned: drainage area, annual average precipitation, minimum elevation, and



Experiment 4: Watershed Delineation

- rcode(VA): string
- xlocation(lon): float
- ylocation(lat): float
- crs(4326): int
- includeparameters(false): boolean
- includeflowtypes(false): boolean
- includefeatures(true): boolean
- simplify(ture): boolean

Experiment 4: Basin Characteristics

- rcode(VA): str
- workspaceID(): string
- includeparameters(DRNAREA,PRECIP, LC11IMP,MINBELEV,LC11DEV, LC11WATER,LC11WETLND, LC11FORSHB,LC11CRPHAY, LC11GRASS,LC11BARE): string

Figure 7 | URL parameters used for the web services tested in Experiment 4.

National Land Cover Database 2011 land cover percentages. These characteristics were chosen using the includeparameters setting. The URL for the watershed delineation service was written, requested, and returned before the same process was initiated for the basin characteristics service so that only one service request was in process at any given time (Figure 8). The URLs were called 25 times.

Experiment 5: location of server and client machines

VMs in three different geographic locations, Northern Virginia, Oregon, and Northern California, were used in this experiment as the client machine for making the service requests. Scripts for the USDA, EPA WATERS, and StreamStats services were started at the same time in the three different locations. Information about the hardware used

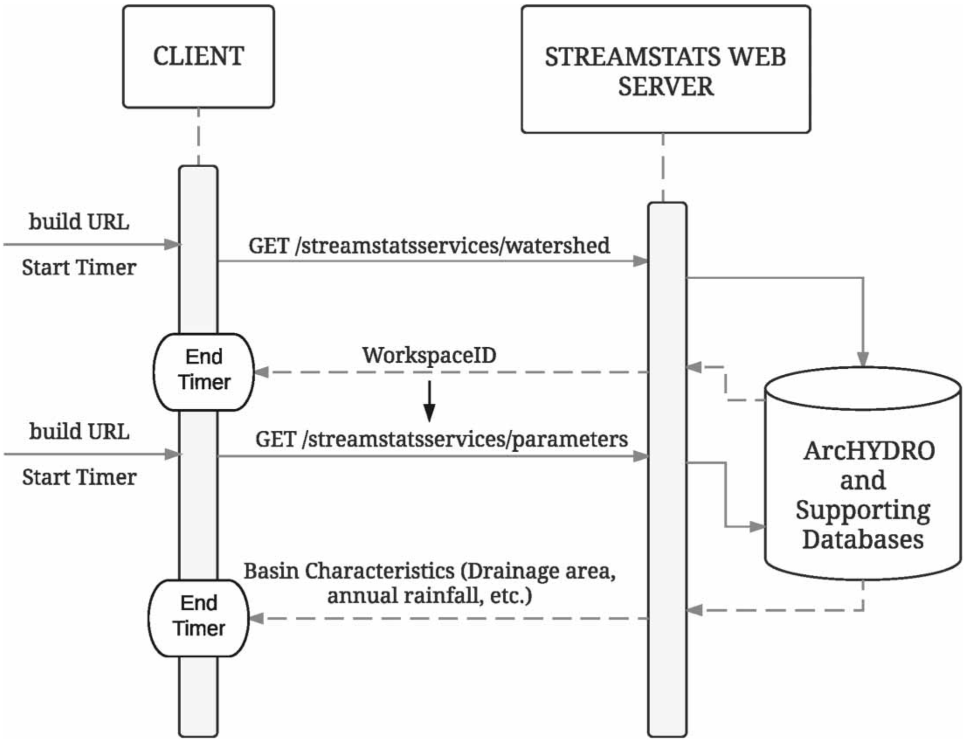

CLIENT
STREAMSTATS WEB SERVER
build URL
Start Timer
GET /streamstatsservices/watershed
End Timer
WorkspaceID
build URL
Start Timer
GET /streamstatsservices/parameters
ArcHYDRO and Supporting Databases
Basin Characteristics (Drainage area, annual rainfall, etc.)
End Timer

Figure 8 | Sequence diagram for the web services tested in Experiment 4.

by these web providers such as server locations and specifications, are confidential and cannot be provided by the sponsored agency. These scripts were in the same form as the first four experiments. The URL parameters were set to general values and kept constant so that only the client location was varied for this experiment. The Soils Data Access service was bounded by coordinates representing an approximately 25,000 acre polygon. All other parameters remained the same as in Experiment 1. EPA WATERS was given the same starting location as Experiment 3. The downstream search distance was set to 25 km, the navigation type to DM, and the 303(d) event list was populated for all trials. Finally, the StreamStats watershed was the 2,500 acre watershed tested in Experiment 4. The sequence for this experiment follows that of the first four experiments. The Soils Data Access, WATERS, and StreamStats URLs were requested 25 times each to quantify variability in response time. Three VMs were initiated, one in each geographic location, and Python scripts for the individual service to be tested were loaded onto the VMs local directory. In order to minimize the effect of network traffic differences at the three locations, all tests in Experiment 5 were run at approximately 1 p.m. eastern time during a workday. Results were stored separately and copied into a single file after the scripts had completed.

RESULTS

The results of Experiment 1 showed how the requested polygon size affects the response times for the Soils Data Access web services (Figure 9). For polygons 1,000 acres and smaller, the requested information was returned in under 1 second for both of the layers requested. The response time was approximately three times slower for the mapunitpoly service and ten times slower for the mapunitpolyextended service when requesting data for a 10,000 acre polygon compared to the 1, 10, and 100 acre polygons.

A one-way ANOVA showed that the polygon size did have a significant effect on response times for the mapunitpoly layer when comparing all five polygon sizes at the $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 1897.23 > 2.45$, $p \sim 0$]. Post hoc comparisons indicated that the 1, 1,000, and 10,000 acre polygon sizes were significantly different from the others, but that the 10 and 100 acre polygon response times did not significantly differ. There was also a significant effect of polygon size on response times for the mapunitpolyextended layer at the $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 612.84 > 2.45$, $p \sim 0$]. Post hoc comparisons indicated that the 1, 10, and 100 acre polygon response times were not significantly different from one another, while the 1,000 and 10,000 acre polygons were significantly different.

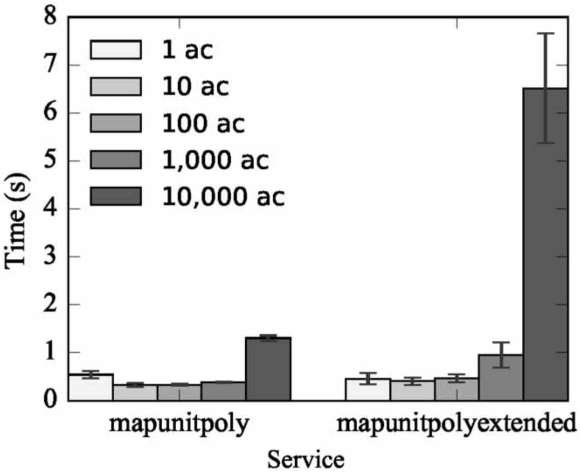


Figure 9 | Average response times and standard deviations from Experiment 1.

The results of Experiment 2 show how the requested starting location, specifically the distance away from the stream network, affects response times for the EPA WATERS' web services (Figure 10). There was

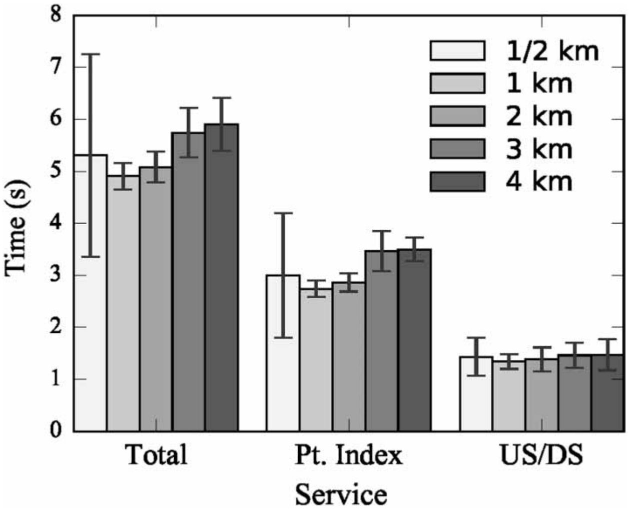


Figure 10 | Average response times and standard deviations from Experiment 2.

approximately a 25% increase in response times for points 3–4 km away from the stream network when compared to the points 0.5–2 km from the stream network. The point indexing service took 1–2 seconds longer than the upstream/downstream search service. A one-way ANOVA used to compare the effect of initial distance from a flowline on response times for WATERS services showed there was no significant effect of starting coordinates on the response times for the upstream/downstream search service at the $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 1.04 < 2.45$, $p < 0.389$]. The upstream/downstream search service returned times just under 1.5 seconds. There was a significant effect of starting location on response times for the point indexing service at the $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 8.59 > 2.45$, $p \sim 0$]. Post hoc tests indicated that the 0.5, 1, and 2 km distances were not significantly different from one another, and that the 3 and 4 km distances were not significantly different from one another.

The results of Experiment 3 show how the requested downstream search length affects response times for the EPA WATERS web services (Figure 11). The 50 km search distances took approximately seven times longer to return compared to the 1 km search distances. The ANOVA indicated that the response times for the point indexing service were not significantly different at a $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 0.26 < 2.45$, $p < 0.901$]. However, there was a significant effect of search distance on response for the upstream/downstream search service at a $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 56.66$, $p \sim 0$]. Post hoc tests indicated that the 1 and 5 km response times were not significantly different from each other and that the 25 and 50 km response times were not significantly different from each other.

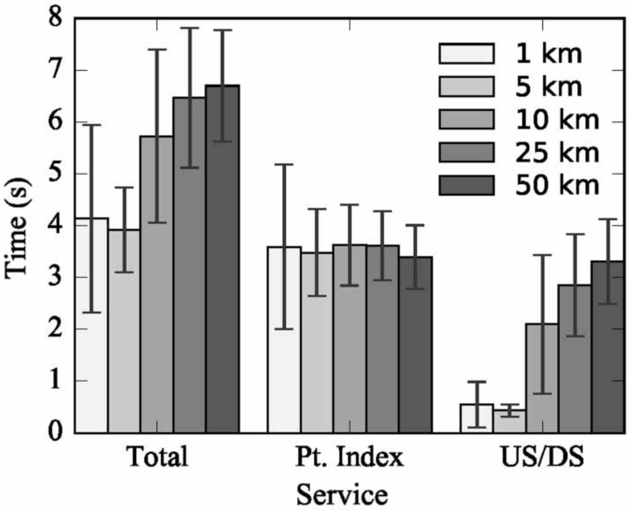


Figure 11 | Average response times and standard deviations from Experiment 3.

The upstream/downstream search service starts at coordinates on a flowline and then travels a specified distance downstream and outputs events that are encountered, in this case any 303(d) listings. To determine if response times were due to more events being encountered and therefore the message size increasing, the response message size was also recorded (Table 3). Internet speeds were on the order of 10–100 Mbits/second for the t2.micro machine. The approximately 54 kB increase in message size between the 1 and 50 km search distance would only justify an increase of 0.005 to 0.05 seconds in response time, thus the increasing message size was not a significant cause for the increased response time.

The results of Experiment 4 show how the size of the delineated watershed affects response times for the StreamStats web services (Figure 12). Response times were relatively constant despite the fact that the watershed size increased substantially. This was validated with the ANOVA test at a $p < 0.05$ level [$F_{crit}(4,120) = 2.45$, $F(4,120) = 0.493 < 2.45$, $p < 0.741$] for the watershed delineation service as well as for the basin characteristics service [$F_{crit}(4,120) = 2.45$, $F(4,120) = 1.784 < 2.45$, $p < 0.137$]. The watershed delineation responded in around 40 seconds while the basin characteristics were returned within 15–20 seconds for all watershed sizes tested. Despite scaling well

Table 3 | Returned file sizes for Experiment 3

Service	Search distance	File size (kB)
Point indexing	1	1.5
	5	1.5
	10	1.5
	25	1.5
	50	1.5
Upstream/downstream	1	5.3
	5	8.6
	10	11.6
	25	34.1
	50	59.2

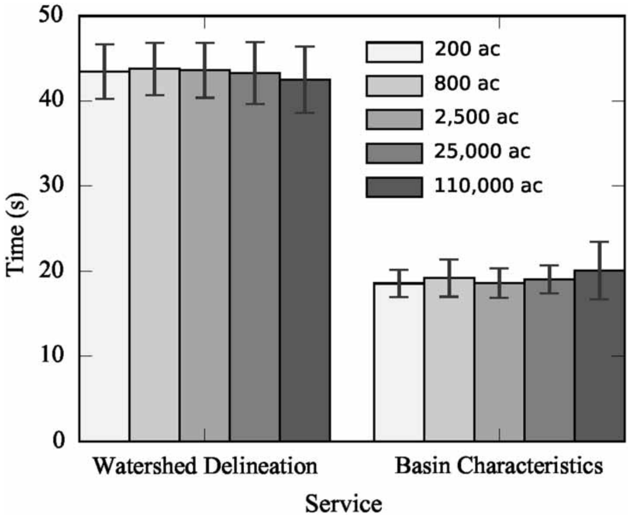


Figure 12 | Average response times and standard deviations from Experiment 4.

to increasing study area sizes, these services were the slowest of the ones tested by a factor of nearly ten.

The results of Experiment 5 show how the client's location affects response times for all service providers tested (Figure 13). ANOVA tests indicated that all services tested were significantly affected by the client location at the $p < 0.05$ level. The Northern Virginia location had longer response times for the Soils Data Access services and for the StreamStats services when compared to the two western US locations. However, it had shorter response times compared to the two western locations for the WATERS services. Post hoc testing indicated that the two locations in the western United States were not significantly different for most services, except for the soils services, with the Northern California location returning responses more quickly than the Oregon location.

DISCUSSION

Performance of services

To provide some context for the response times for each of these services, Nielsen (1994) suggests three thresholds for web application response times. At and below 0.1 seconds, the user feels that the application is reacting instantaneously. One second represents the limit to keep user's attention uninterrupted, although the delay is noticeable. Ten seconds is the limit for keeping a user's attention on the application. Returns longer than 10 seconds should have some type of progress icon or offer asynchronous capabilities. In a more recent update, Nielsen (2014) stands by his usability recommendations as they are based on user experience and not the performance of an application.

It is important to keep in mind that these services are automating complex and tedious tasks that, when traditionally performed with desktop computers, can take several minutes and even hours to complete depending on the size of the data and number of processing steps required. Thus, while 10 seconds may seem very quick for performing

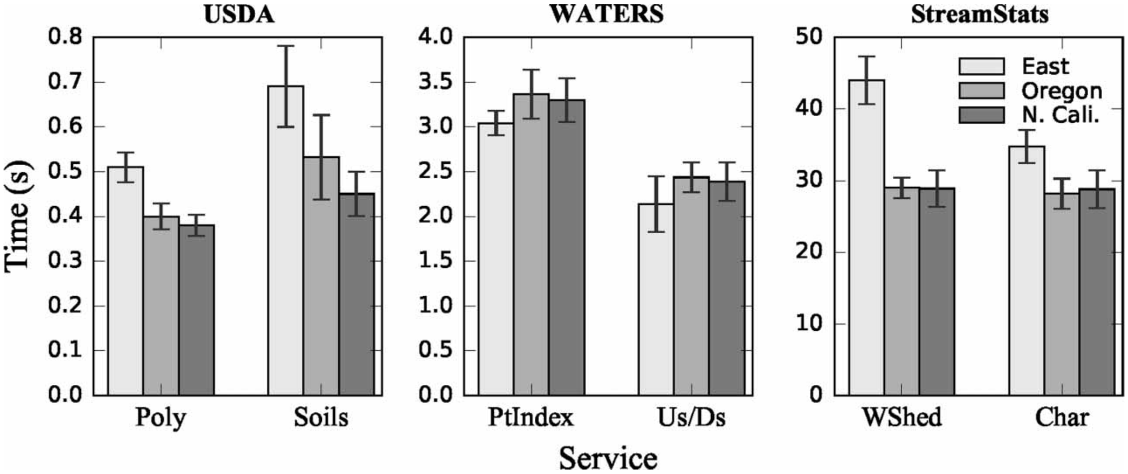


Figure 13 | Average response times and standard deviations from Experiment 5.

a task like watershed delineation, in a distributed, on-demand DSS, having a response within 10 seconds would allow for a more interactive user experience compared to waiting longer for the response. It is also important to be clear that not all environmental DSSs will require on-demand, instantaneous responses. For example, groundwater management DSSs (Le Page *et al.* 2012) may not often require instantaneous responses because waiting several minutes or even an hour for on demand data access and integration will not impede decisions based on the model results. However, the availability of on demand systems can have a significant benefit for DSSs that support more dynamic systems such as reservoir operation, pump station operation and flood warning systems (Abebe & Price 2005; Savić *et al.* 2011).

Based on the study results, the two Soils Data Access web services had the shortest response times. In most cases, except for the 10,000 acre study area, both services returned before the 1 second threshold. The mapunitpoly service consistently returned faster than the mapunitpolyextended service, with the difference increasing as the size of the polygon increased. The two WATERS services tested were the next fastest services to return. The point indexing service was slower than the upstream/downstream search service. However, as the downstream search distance increases, the two services' response times approached each other. The two StreamStats services were the slowest to return of the ones tested with both being over the 10 second threshold for the watershed sizes tested.

To reiterate, the experiments were designed to attempt to identify parts of the services where potential bottlenecks may occur, along with general response times for the services. The most obvious impediment to an on-demand DSS which relies on the services tested would be the response times for the StreamStats web services. However, these services are also performing arguably the most significant data processing before returning information to the user. Innovative methods to speed up these services, such as new watershed delineation algorithms, would benefit on-demand applications. Another area for improvement would be for soil data requests for large areas. The Soils Data Access mapunitpolyextended service took six times longer to obtain soils characteristics for 10,000 acre study areas when compared to 1,000 acre study areas. For the application that motivated this study, projects would most likely be less than 1,000 acres, however other use cases may require soil data for larger areas. Although the initial distance away from the stream network does affect response times, the effect is small relative to other response times. However, the search distance downstream should be considered within a DSS utilizing these services. The response times were almost seven times longer for a 50 km search distance when compared to the 1 km search distance. Therefore, it may be necessary to maintain a maximum downstream search distance that users cannot exceed.

For all six web services, but especially the two provided by StreamStats, there was a significant difference between response times from client VM's located in different parts of the country. This result suggests that efforts to direct service requests to more nearby servers could have a significant impact on response times. Other factors such as network traffic may also have played a role in these results, however, and further testing could better pinpoint the exact cause for these delayed response times. What can be taken away from the experiment is that the client's location affects web service response times in a somewhat surprisingly significant way. For example, the watershed delineation service offered by StreamStats experienced a 34% reduction in response times when the client was located in the Western United States compared to the Eastern United States (Figure 13).

Variability and reliability of services

StreamStats services' standard deviations exceeded 1 second in all cases (Figures 12 and 13). However, when comparing the average coefficients of variation (CV), where the standard deviation is normalized by the mean, the StreamStats services were the least variable with an average CV of 0.13. WATERS services' deviations ranged between 0.2 and 1 second (Figures 10, 11 and 13) with an average CV of 0.22. Soils Data Access services were consistently under 0.2 seconds for most trials (Figures 9 and 13) with an average CV of 0.17.

Reliability concerns that affect DSS implementation include timeouts and no data or wrong data returns. In addition to the requests made for the actual experiments, these services were also requested many times during setup and code debugging. In total, hundreds of requests were made to each of the services and no timeouts or data errors were encountered. StreamStats is under active development with the USGS rolling out updates for each state and beta versions being tested, with one just being made available publically. In the past, WATERS has retired services that were considered obsolete or unpopular. This could become a problem if distributed DSSs depend on services for reproducing past studies and better ways to maintain or archive legacy services is an important area for future work. The Soils Data Access is taken offline for maintenance from 12 a.m. to 4 a.m. Mountain Time every night in order to complete updates. Overall, although no issues were specifically encountered during testing, these are factors that need to be considered before using these web services within production systems.

Remaining barriers to achieving the on-demand environmental DSS

In regard to an environmental DSS to support the stormwater management application, there are several areas that need attention. First, there is required data that is unavailable or not easily obtained through web services. We were unable to find an authoritative web service for land cover data. StreamStats does provide some land cover derived data within its basin characteristics service, however this is not consistent for all states.

Other data needed for the scenario were more localized data such as the locations of nutrient credit banks, amount of available credits, and pricing. Although federal data providers are making increased use of web services and their adoption at local governmental levels is growing for standardized data like geospatial data layers, more specialized datasets without well-established standards are still a work in progress. For example, nutrient credit pricing is localized information that would be ideal for access through a web service. Prices may change frequently, but as long as the information was updated by the service provider, the consumer (in this case a project manager interested in purchasing credits) could make a request using web services and be provided with current and accurate information.

Another area for improvement is to the service response times. The USDA web services are suitable for use within a distributed DSS because their response times were consistently below 1 second for most tests. It is only when the study area becomes very large (10,000 acres in our analysis) that the service slows to a point where on-demand access is no longer practical. The WATERS services returned responses within 5–6 seconds typically, except in the cases where the downstream search length was below 5 km or above 10 km. These response times could also be acceptable for on-demand access, especially if there is a progress bar for users. StreamStats response times were too long for an on-demand application (often over 40 seconds for the watershed delineation). Although StreamStats services are performing complex calculations that would take significant time if performed manually, they are not yet at the point where on-demand applications can make use of the services without an effort to make sure users stay engaged during the waiting time (Nielsen 1994).

In order to create an environmental DSS using on-demand web services, the response times for the services should improve and more attention should be paid to methods for speeding up complicated analysis services like those provided through StreamStats or Soils Data Access for large areas. As the use and demand for these services continues to grow, organizations will continue to invest in making the services more responsive and user friendly. One approach for improving response times may be simply to run the services on larger physical servers. Another approach would be to take advantage of recent advances in cloud computing services that allow for dynamic scaling. Dynamic scaling can automatically provision resources for web applications based on demand, allowing services to be highly responsive without large upfront resource investments. Other enhancements, such as replicating web services at different geographical locations or reducing message sizes between clients and servers, could significantly improve response times. For example, Experiment 5 demonstrated that the location of the client compared to the server resulted in a consistent 10–20% difference in response times and in some cases as much as a 50% difference in response time.

CONCLUSIONS

This research explored the feasibility of using available web services from federal agencies to support a distributed environmental DSS with on-demand data access. The popularity of web applications has given rise to web services as a new area for information dissemination due to the benefits offered in interoperability, access, and standardization. Previous work has been done on using web services for data access, analysis, visualization, and simulation of environmental processes. Most of the work has been focused on the design or implementation of the services themselves. There has been less work on examining the services' performance for building a distributed environmental DSS application. A stormwater management application was used to define typical service requests for an environmental DSS. Experiments were designed to test potential bottlenecks for service requests from the three major federal agencies.

The Soils Data Access services averaged response times below 1 second for study areas 1,000 acres and below. The WATERS services responded between 4 and 7 seconds. The initial distance from the stream network did not impact response times as much as the downstream search distance. The StreamStats services responded in 40 seconds on average for basin delineations and between 15 and 30 seconds for the basin characteristics, depending on the number of parameters requested. The StreamStats service response times were relatively constant for increasing watershed sizes. Client location did have an effect on the response times for all three services, generally with a difference of 12–25% and as much as 34% in response times.

The variability in response time for the same service call repeated at different times ranged from less than a tenth of a second to a few seconds, depending on the service. CV for the experiment trials ranged between 5% and 70%, but were generally below 30%. Reliability concerns stemming from timeouts or requests not being returned were not encountered. The USDA web services are already suitable for a distributed environmental DSS due to their short response times for typical project study areas. The WATERS services are suitable as well, however additional improvement in response times would be beneficial in order to reduce response times below 1 second. The StreamStats services are less suitable for a distributed DSS application due to long response times of over 40 seconds. When using the StreamStats service for watershed delineation and characterization, a progress bar or asynchronous communication would be necessary to keep users engaged.

Although the services tested are making progress toward the long term vision of distributed environmental DSSs, there are opportunities for future research and development. Future work should be devoted to creating new web services needed for environmental DSS applications. For example, services for land use and land cover data and more localized information are needed. There is also a need for improvement to the algorithms supporting more rapid watershed delineation, especially for small watersheds. Web services providers may also benefit from geographically distributing their services or reducing message sizes in order to handle requests from different parts of the country more quickly, since this was also found to be a significant factor in service response time. With these further advancements, web services will be able to better fulfill the longer-term vision of distributed DSSs with on-demand access to data, analysis, and visualization routines.

ACKNOWLEDGEMENTS

This research was funded in part through a grant from the Mid-Atlantic Transportation Sustainability Center (MATS) University Transportation Center (UTC). Roy Mills and John Olenick of the Virginia Department of Transportation (VDOT) and Mike Fitch of the Virginia Transportation Research Council (VTRC) assisted in defining the stormwater management application.

REFERENCES

Abebe, A. J. & Price, R. K. 2005 Decision support system for urban flood management. *J. Hydroinform.* **7** (1), 3–15.

Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T. & Valentine, D. 2012 Hydrodesktop: web services-based software for hydrologic data discovery, download, visualization, and analysis. *Environ. Model. Softw.* **37**, 146–156.

Blodgett, D., Lucido, J. & Kreft, J. 2016 Progress on water data integration and distribution: a summary of select US geological survey data systems. *J. Hydroinform.* **18** (2), 226–237.

Booth, N. L., Everman, E. J., Kuo, I. L., Sprague, L. & Murphy, L. 2011 A web-based decision support system for assessing regional water – quality conditions and management actions. *JAWRA J. Am. Water Resour. Assoc.* **47** (5), 1136–1150.

Brodaric, B. & Piasecki, M. 2016 Water data networks: foundations, technologies and systems, implementations, and uses. *J. Hydroinform.* **18** (2), 149–151.

Buytaert, W., Baez, S., Bustamante, M. & Dewulf, A. 2012 Web-based environmental simulation: bridging the gap between scientific modeling and decision-making. *Environ. Sci. Technol.* **46**, 1971–1976.

Chen, C. P. & Zhang, C. Y. 2014 Data-intensive applications, challenges, techniques and technologies: a survey on big data. *Inform. Sci.* **275**, 314–347.

Chen, Y. & Han, D. 2016 Big data and hydroinformatics. *J. Hydroinform.* **18** (4), 599–614.

Choi, J. Y., Engel, B. A. & Farnsworth, R. L. 2005 Web-based GIS and spatial decision support system for watershed management. *J. Hydroinform.* **7** (3), 165–174.

Díaz, L., Granell, C. & Gould, M. 2008 Case Study: geospatial processing services for web-based hydrological applications. In: *Geospatial Services and Applications for the Internet* (J. T. Sample, K. Shaw, S. Tu & M. Abdelguerfi, eds). Springer, New York, pp. 31–47.

Environmental Protection Agency 2015 WATERS web services. Retrieved from: www.epa.gov/waterdata/waters-web-services (accessed July 2017).

Feng, M., Liu, S., Euliss, N. H., Young, C. & Mushet, D. M. 2011 Prototyping an online wetland ecosystem services model using open model sharing standards. *Environ. Model. Softw.* **26** (4), 458–468.

Fielding, R. T. 2000 *Architectural Styles and Design of Network-Based Software Architectures*. Doctoral dissertation, University of California, Irvine.

Galdiero, E., De Paola, F., Fontana, N., Giugni, M. & Savic, D. 2016 Decision support system for the optimal design of district metered areas. *J. Hydroinform.* **18** (1), 49–61.

Goodall, J. L., Horsburgh, J. S., Whiteaker, T. L., Maidment, D. R. & Zaslavsky, I. 2008 A first approach to web services for the national water information system. *Environ. Model. Softw.* **23** (4), 404–411.

Goodall, J. L., Robinson, B. F. & Castronova, A. M. 2011 Modeling water resource systems using a service-oriented computing paradigm. *Environ. Model. Softw.* **26** (5), 573–582.

Granell, C., Díaz, L. & Gould, M. 2010 Service-oriented applications for environmental models: reusable geospatial services. *Environ. Model. Softw.* **25** (2), 182–198.

Harvey, H., Hall, J. & Peppé, R. 2012 Computational decision analysis for flood risk management in an uncertain future. *J. Hydroinform.* **14** (3), 537–561.

Hey, T. & Trefethen, A. E. 2005 Cyberinfrastructure for e-Science. *Science* **308** (5723), 817–821.

Horsburgh, J. S., Tarboton, D. G., Piasecki, M., Maidment, D. R., Zaslavsky, I., Valentine, D. & Whitenack, T. 2009 An integrated system for publishing environmental observations data. *Environ. Model. Softw.* **24** (8), 879–888.

Horsburgh, J. S., Tarboton, D. G., Schreuders, K. A., Maidment, D. R., Zaslavsky, I. & Valentine, D. 2010 HydroServer: A platform for publishing space-time hydrologic datasets. In: *Proceedings of the AWRA Spring Specialty Conference on GIS and Water Resources*, March 29–31, Orlando, FL.

Kumar, S., Godrej, A. N. & Grizzard, T. J. 2015 A web-based environmental decision support system for legacy models. *J. Hydroinform.* **17** (6), 874–890.

Laniak, G. F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M. & Peckham, S. 2013 Integrated environmental modeling: a vision and roadmap for the future. *Environ. Model. Softw.* **39**, 3–23.

Le Page, M., Berjamy, B., Fakir, Y., Bourgin, F., Jarlan, L., Abourida, A., Benrhanem, M., Jacob, G., Huber, M., Sghrer, F. & Simonneaux, V. 2012 An integrated DSS for groundwater management based on remote sensing. The case of a semi-arid aquifer in Morocco. *Water Resour. Manage.* **26** (11), 3209–3230.

Lu, B. & Piasecki, M. 2012 Community modeling systems: classification and relevance to hydrologic modeling. *J. Hydroinform.* **14** (4), 840–856.

Maidment, D. R. 2002 *Arc Hydro: GIS for Water Resources.* ESRI Press, Redlands, CA.

Matthies, M., Giupponi, C. & Ostendorf, B. 2007 Environmental decision support systems: current issues, methods and tools. *Environ. Model. Softw.* **22** (2), 123–127.

Mineter, M. J., Jarvis, C. H. & Dowers, S. 2003 From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. *Environ. Model. Softw.* **18** (4), 379–391.

Nielsen, J. 1994 *Usability Engineering.* Morgan Kaufmann, San Francisco, CA.

Nielsen, J. 2014 Response Times: The 3 Important Limits. Nielsen Norman Group. Retrieved from www.nngroup.com/articles/response-times-3-important-limits/, (accessed July 2017).

Quiroga, V. M., Popescu, I. A., Solomatine, D. P. & Bociort, L. 2013 Cloud and cluster computing in uncertainty analysis of integrated flood models. *J. Hydroinform.* **15** (1), 55–70.

R Development Core Team 2008 *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. Retrieved from www.R-project.org.

Savić, D. A., Bicik, J. & Morley, M. S. 2011 A DSS generator for multiobjective optimisation of spreadsheet-based models. *Environ. Model. Softw.* **26** (5), 551–561.

Sun, A. 2013 Enabling collaborative decision-making in watershed management using cloud-computing services. *Environ. Model. Softw.* **41**, 93–97.

Szalay, A. & Gray, J. 2006 2020 computing: science in an exponential world. *Nature* **440** (7083), 413–414.

Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodall, J. L., Band, L., Merwade, V., Couch, A., Arrigo, J. & Hooper, R. 2014 HydroShare: Advancing Collaboration Through Hydrologic Data and Model Sharing. In: *Proceedings of the 7th International Congress on Environmental Modelling and Software.* International

Environmental Modelling and Software Society, San Diego, CA, pp. 978–988.

Van Griensven, A., Breuer, L., Di Luzio, M., Vandenberghe, V., Goethals, P., Meixner, T., Arnold, J. & Srinivasan, R. 2006 Environmental and ecological hydroinformatics to support the implementation of the European water framework directive for river basin management. *J. Hydroinform.* **8** (4), 239–252.

Wagener, T., Reed, P., van Werkhoven, K., Tang, Y. & Zhang, Z. 2009 Advances in the identification and evaluation of complex environmental systems models. *J. Hydroinform.* **11** (3–4), 266–281.

Walker, J. D. & Chapra, S. C. 2014 A client-side web application for interactive environmental simulation modeling. *Environ. Model. Softw.* **55**, 49–60.

First received 14 July 2017; accepted in revised form 30 October 2017. Available online 29 November 2017