7th ISE & 8th HIC
Chile, 2009

SUBMISSION AND QUERYING TOOLS FOR A HYDROLOGIC INFORMATION SYSTEMS DATABASE

KEVIN MCCARTY
University of Idaho, Dept. of Computer Science,1776 Science Center Dr. Idaho Falls, ID 83402 USA
Tel:00-1-208-282-7933, email: thekev@cableone.net

MILOS MANIC
University of Idaho, Dept. of Computer Science,1776 Science Center Dr. Idaho Falls, ID 83402 USA
Tel:00-1-208-282-7933, email: misko@uidaho.edu

PETER GOODWIN
University of Idaho, Center for Ecohydraulics Research, 322 E. Front St., Ste. 340, Boise, ID 83702 USA
Tel: 00-1-208-364-4081, email: pgoodwin@uidaho.edu

MICHAEL PIASECKI
Drexel University, Dept. of Civil Engineering, 3141 Chestnut Street, Philadelphia, PA, 19104 USA
Tel: (215) 895-1721; FAX: (215) 895-1363, email: Michael.Piasecki@drexel.edu

ABSTRACT
The recent establishment of the WATERs information network in the US, has prompted a number of entities to joint this network beyond the initial selected set of test bed nodes. The state of Idaho is supporting the creation of a IdahoWaters node through its EPSCoR program with them aim of not only providing a single access point for Idaho water information but also to make these data holdings accessible nationwide through participation in the network. Given the many individuals institutions that will participate in this effort, means of data submission are an extremely important aspect when developing an information node of this type. This paper demonstrates an architecture for the submission as well as querying and presentation of large datasets of hydrologic data via the internet. Discussed are the necessary hardware and software configurations used to create databases for staging, permanent storage, online analytical processing and distribution. In addition software and tools for decision support as well as automation for data extraction, transformation and loading are presented. Finally application of this architecture is shown for a wide-scale, distributed, hydrologic-based, collaborative information network.

Keywords: Adaptive architecture, heuristics, modeling, prediction

INTRODUCTION

The CUAHSI Hydrologic Information System http://his.cuahsi.org/ is a cooperative effort whose goal is to provide access for the scientific community to hydrological information along with tools and other resources for data visualization and analysis. CUASHI partners include major government, private and academic institutions in both the United States and abroad. One such node, called IdahoWaters, has been set up for the state of Idaho that seeks to become a central clearing house for all water data in the state. It has een established as part of the Idaho EPSCoR program and currently hosts 2 major watershed data sets, i.e that of Dry Creek (a local watershed in the foothills of Boise, ID) and Reynolds Creek a much larger watershed in the southwest corner of Idaho. The latter is arguably one of the best and most intensly instrumented watersheds in the continental US and as such provides an unprecedented wealth of data. This node is slated to grow over the years, which requires more watershed data to be hosted in the server which originates from a number of academic and governmental entities. This posts the challenge of permitting access to a potentially larger number of outside-the-network data managers that need to submit their data to the node in a network environment that is quite restricted.

The existing system for data gathering and distribution, currently hosted in the Center of EcoHydraulics Research at the University of Idaho consists of a university server, accessible through a virtual private network and internal intranet. The university maintains an Active Directory network, accessible from the outside using Cisco VPN software. The ODM databases reside on a Windows 2003 Server running SQL Server 2005 as shown in Fig. 1.

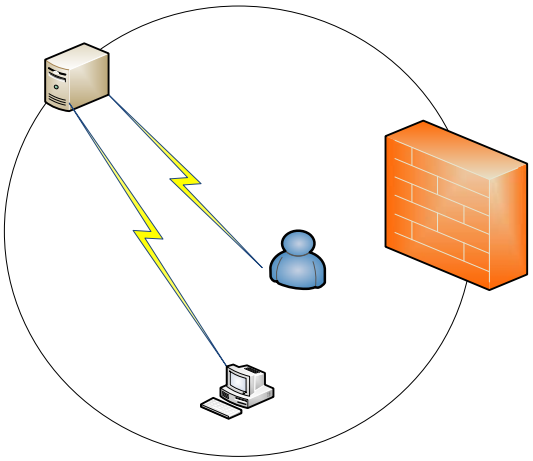

Fig. 1 – ODM Direct Access via VPN

In the existing system, partners can upload or download data using special tools provided. The data is provided in a pre-defined format, usually as an Excel spreadsheet or CSV or tab-delimited file. The user must login to the VPN and then provide a SQL password for access to the database in order to initiate an upload. For downloads, the process is much the same in reverse. For security purposes, the ODM database is not accessible outside the VPN and accounts are required both at the network level as well as the database, although the database account is a general purpose account. Web services do provide some web-based data access to internet-based clients.

The existing system supports a limited number of users and sites but is deemed inadequate to scale to the larger scientific community. A proposed new system architecture was sought, one that would incorporate the existing system and maintain adequate levels of system security but be accessible to the large internet community both for the upload process as well as the data retrieval process [1], [5], [6]. This paper presents a prototype system, implemented as part of a grant from the Idaho NSF EPSCoR RII: Idaho Experimental Watershed Network and a long-range proposal to address this issue. The paper is organized in the following sections: Section II presents the problem statement. Section III discusses the proof-of-concept architecture and proposed solution. Section IV presents tests results. Section V presents the conclusion and future work.

PROBLEM STATEMENT

The current ODM implementation team has identified a number of shortcomings in the existing system, particularly with regards to data input:

1. The current system requires direct access to the network. This presents an administrative problem with the larger scientific community as they are unlikely to wish to create network accounts and the internal network administrative team as they would have to maintain a large number of non-affiliated users. The security implications of such an implementation are enormous and considered impractical for this purpose [5], [6]. Because of the diverse locations and potential base of users, any solution has to make use of the World Wide Web in order to ensure widespread acceptance and usage [9].
2. SQL Accounts are being shared. As with issue #1, the security implications are enormous and impractical for a large user base. This is not a best practice, both from a security standpoint and from an enterprise architecture standpoint as it creates multiple points of entry into the system as well multiple transaction paths [5], [6].
3. Database access is direct. This is also not a best practice as the potential for mistakes from users or mischief from hackers is quite high [5].
4. Only a limited number of upload formats are permitted. Many formats provided by existing systems would require an additional conversion or otherwise be unsupported [8].
5. Uploads are initiated via a user-driven process.
6. Rollbacks and recovery, audit trails and history are limited.

7. There are no mechanisms outside of the Data Loader and simple database keys to ensure data integrity and conformance to standards.

PROOF-OF-CONCEPT AND PROPOSED SOLUTION

Because issue #1 was the overriding concern, the prototype had to be one which allowed data to be transferred via the web to the database server without using the existing network/VPN access. In addition as much automation as possible was to be implemented to ensure a hands-off transfer of data to the largest extent possible and the groundwork laid for addressing issues 2-7. The prototype configuration involved renting a staging server, Internet IP address and bandwidth from an e-Business provider and installation of a copy of Microsoft's SQL Server 2005. The prototype configuration was implemented as shown in Fig. 2.

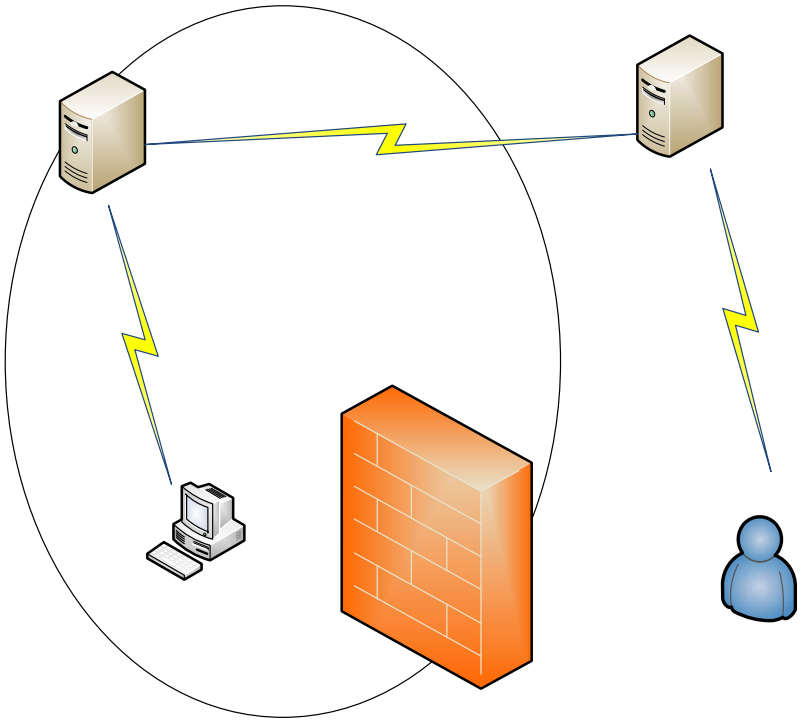

Fig 2. Prototype Configuration

The prototype used the existing configuration in addition to the staging server. The original ODM Server remains as such, but outside access is no longer permitted for uploads. In both the ODM server as well as the staging server, additional components of SQL Server, included with the commercial software, were installed. On the ODM server: SQL Server Analysis Services (SSAS), SQL Server Integration Services (SSIS) and on the staging server: SSIS.

SQL Server 2005 Integration Services is the software upgrade to Data Transformation Services (DTS) which was supplied with Microsoft's SQL Server 2000. DTS was a maintenance and automation tool for SQL Server, used to perform repetitive tasks such as dataloads, system maintenance and data transformation [8], [9]. SSIS added significant new functionality to DTS in the form of enhanced control logic, programmability, added features, logging, and events. Not only could SSIS packages be developed to import and export a large number of dataset types, such as Excel, Text, XML, Oracle and others, a package could be called from an external application such as a web page or application program or even a remote SQL Server.

In the prototype system, the Staging Server serves as the external link between the user inputting the data and the ODM databases on the ODM Server. It does so by exposing both an HTTP virtual directory (for WWW clients) and an FTP directory (for internal clients) the users can access to upload datafiles as shown in Fig 3.

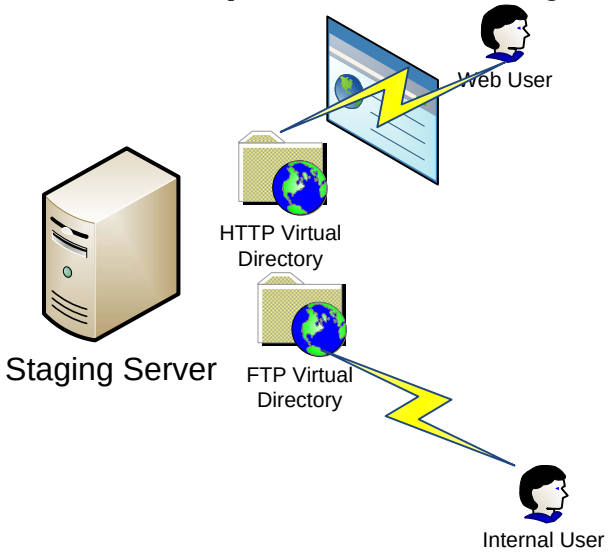

Fig. 3 – Staging Server Configuration

Users now upload files via a web-page or copy them directly via FTP through an application such as Internet Explorer. The web-page then initiates an SSIS package run which imports the file into the database without any assist from the user. In addition, a SQL Agent process regularly scans the FTP directories for new data and automatically initiates an SSIS-based import if new data is found.

SSIS imports the new data into a stage ODM database, performing any necessary data "cleanup" or ETL operations before making the data available to the ODM Server. In the event data is unacceptable, the Staging Server can email the user, if email is provided, of the status of the data as well as write to a log file and record problems and

errors. In addition to serving as an external bridge from the ODM Server to users, the prototype implementation has a number of additional benefits:

1. The ability to notify internal administration and external users of system/data status, as well as automatically respond to errors as data is being loaded.
2. Apply any number of business rules to clean and prepare the data so that import to the ODM Server consists of a data copy only.
3. Routines to archive files and clean directories of data once it is used.
4. Flexibility in how, where and how much data is loaded.

Once the data has been imported into one or more of the stage ODM databases, it writes to a database table parameters which indicate what new data is available and where it is located. At regular intervals, a SQL Agent running on the production ODM Server submits a request for any new data that is available. If data is available, the ODM Server runs an SSIS package to retrieve all the new datasets, importing them into its production databases before issuing commands to clean them off the Stage Server. The overall prototype configuration now completely isolates the production ODM Server while providing a bridge to the internet users and is represented by Fig. 4.

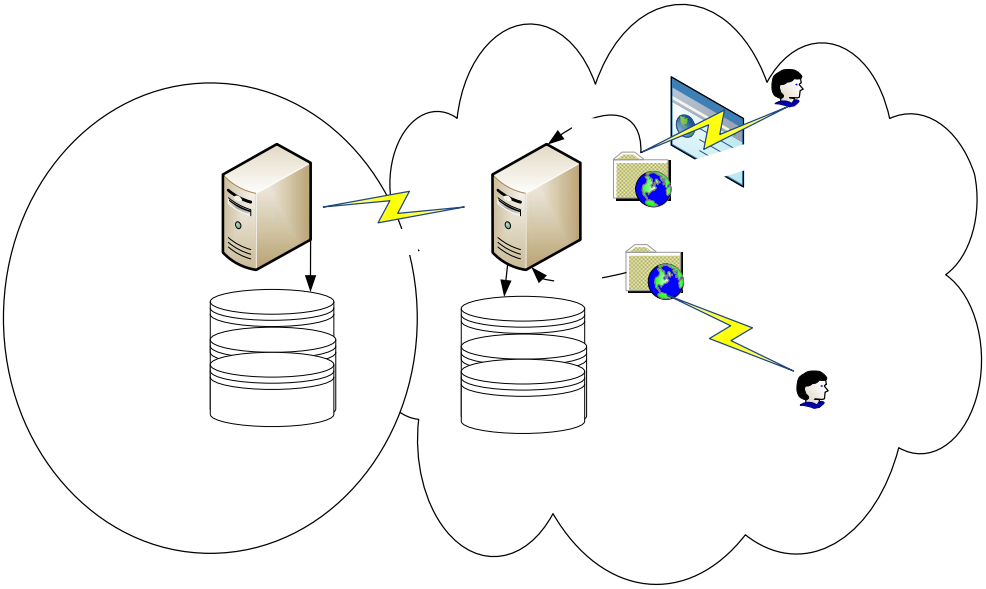

Fig 4. Prototype Configuration

Going beyond the prototype to the final solution will require the integration of a web server, web services and another technology bundled with SQL Server 2005: Service Broker [5], [9]. Service Broker allows two SQL Servers to establish a secure communication channel using HTTP. Whereas the current communication link is only one way (Production Server sends requests to Stage Server) a Service Broker enabled

link will run both ways, allowing direct and secure communications between the ODM Server and Stage Server. This will enable real-time updates from the Stage Server as well as further automate logging and transaction processing. The final proposed solution will be implemented as shown in Fig. 5. The staging area will consist of a SQL Server Farm and support multiple sources of input from automated sensors, users, and even other databases on remote systems. Input will range from flat files to direct database to database replication.

Once data resides in the staging area, it undergoes Extraction, Transformation and Loading (ETL) processes to ensure the data conforms to applicable business rules [1], [2], [3], [7]. From there a Service Broker link moves the data to its final repository which consists of databases along with single and multi-dimensional data-marts organized into a data warehouse. Another Service Broker link using secure TCP endpoints provides the final link to a web farm which provides web services for data access for both web and internal clients. Middle tier objects will exist on both the Production Server in the form of stored procedures and CLR objects and on the Web Server in the form of web services.

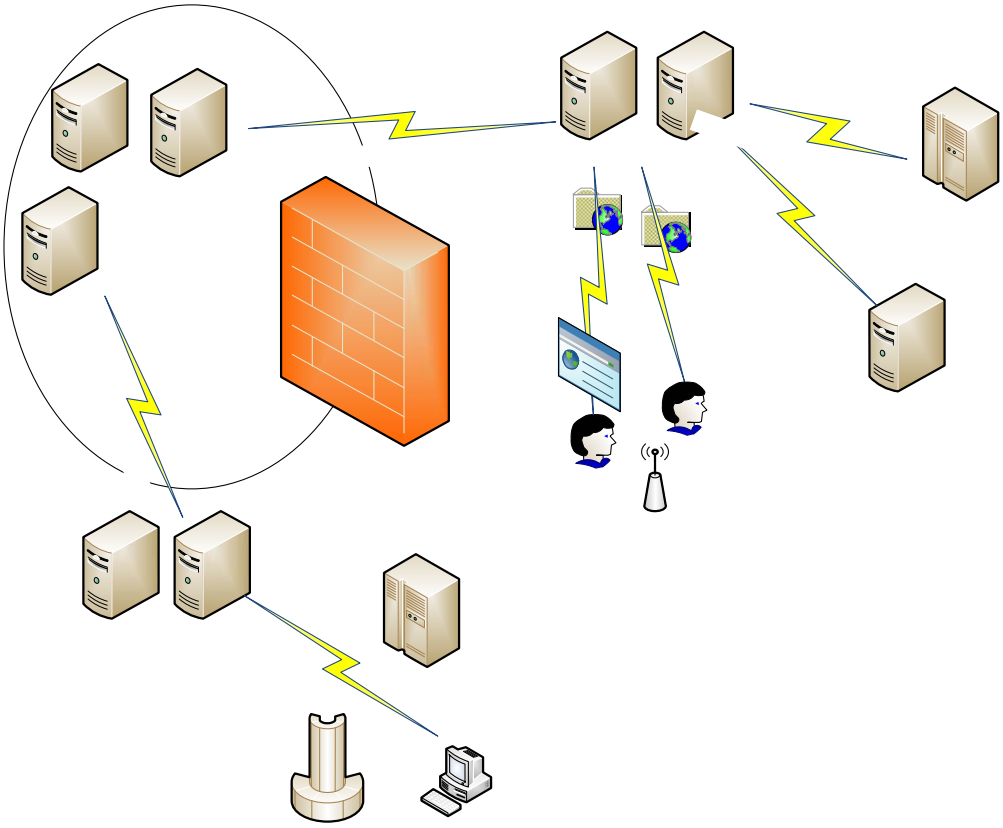

Fig 5. Proposed Solution

TEST RESULTS

A sample ODM dataset was used to generate a text file. Using an FTP connection over the web, the file was uploaded into the virtual FTP directory. Importing the file into the database was done via a batch file, which executed a .NET application program, which executed an SSIS package which performed the initial import. The import data was validated against original source data and the appropriate imports flags and data were verified.

A VPN connection was established with the Production Server. Using Terminal Services to open a session on the Production Server, a command was given to the local SQL Agent to initiate the SSIS package to poll for new data. The package logged into the Stage Server over the web, verified that data existed and sent a download request. Data moved to the production server to the location desired. Data was verified against the original source data.

CONCLUSION AND FUTURE WORK

The prototype demonstrated the ability to have a remote, widely accessible data host without having to compromise the integrity of the secure production server, but it is only one step of many in the implementation. The ODM project is a significant undertaking, with the data component being one of many components required. While the technology is available and proven, tests need to be run to ensure the implementation is sufficiently robust and scalable to meet the anticipated demands of the system. Existing applications need to be modified to use the new architecture. Online Analytical Processing (OLAP) and Data Mining are expected to add significant utility to the overall system but have yet to be explored against the existing datasets.

REFERENCES

[1] J. Han, M. Kamber; *Data Mining Concepts and Techniques*, 2nd Ed, Morgan Kaufmann Publishers, 2006.

[2] Z. Tang, J. MacLennan; *Data Mining with SQL Server 2005*; Wiley Publishing, 2005

[3] S. Harinath, S.R. Quinn, *Professional SQL Server Analysis Services 2005 with MDX*, Wiley Publishing, 2006

[4] I. Ben-Gan, D. Sarka, R. Wolter, *Inside Microsoft SQL Server 2005: T-SQL Programming*, Microsoft Press, 2006

[5] D. Bieniek, M. Hotek, A. Soto, A. Wiernik, R. Dyess, A. Machanic, J. Loria, *SQL Server 2005 Implementation and Maintenance*, Microsoft Press, 2006

[6] B. Beauchemin, *SQL Server 2005 Security Best Practices – Operational and Administrative Tasks*, http://download.microsoft.com/download/8/5/e/85eea4fa-b3bb-4426-97d0-7f7151b2011c/**SQL**2005Sec**Best**Pract.doc, Microsoft White Paper, March 2007

[7] W. McKnight, *Choosing Microsoft SQL Server 2005 for Data* Warehousing, http://www.microsoft.com/sql/techinfo/whitepapers/sql-for-datawarehousing.mspx, Microsoft White Paper, December 2006

[8] B. Knight, A. Mitchell, D. Green, *Professional SQL Server 2005 Integration Services*, Wiley Publishing, January 2006

[9] L. Rubbelke, *Empowering Enterprise Solutions with SQL Server 2005 Enterprise Edition*, http://download.microsoft.com/download/a/c/d/acd8e043-d69b-4f09-bc9e-4168b65aaa71/EmpoweringEnterpriseSolutions.xps, Microsoft White Paper, February 2007