APPLICATION OF PUBLISH/SUBSCRIBE MESSAGING FOR MANAGEMENT OF STREAMING WATER RESOURCE DATA

S. Esswein[1], C. Post[1], J. Hallstrom[2], D. White[3], S. Goasguen[2]

AUTHORS: [1]Dept. of Forestry and Natural Resources, Clemson University, [2]Dept. of Computer Science, Clemson University, Clemson, SC, [3]Clemson University Computing and Information Technology

REFERENCE: Proceedings of the 2008 South Carolina Water Resources Conference, held 14-15, 2008 at the Charleston Area Convention Center, North Charleston, SC

Abstract.

The remote acquisition of data is becoming increasingly commonplace in environmental monitoring applications. The growing demand for remote monitoring is driving the development of new technology and creating new demands for data management solutions. The quantity and diversity of remotely monitored data is growing everyday. Traditional methods of data management fall short in three key areas: supporting real-time data access, dealing with data diversity and supporting high performance & scalability. We present a software solution that addresses these needs without imposing strict requirements at the hardware or application side. This specialized software (also known as middleware) handles the transmission and distribution of structured data and metadata. We leverage the brokered publish/subscribe interaction pattern to provide highly decoupled communications between participating entities. Integrated into this middleware are capabilities to provide real-time processing of observation data for monitoring and quality-assurance/quality control (QA/QC). Additional applications provide a variety of mechanisms for moving observation data onto and off of the middleware.

This software has been implemented in support of the IntelligentRiver™ project and is currently managing observation streams from several monitoring projects. This discussion illustrates its application to water resource monitoring and also provides test results for benchmarking purposes.

Introduction.

The management of streaming observation data poses a significant challenge to large environmental observing systems. These challenges stem from the heterogeneity of producers and consumers, the quantity of streams and the sheer volume and diversity of possible observation parameters. As the size and complexity of these systems grow, so does the need for a scalable, reliable and high performance solution for managing these data streams. This discussion introduces a software solution that supports these needs.

Systems for environmental observing are multifaceted, requiring a technology solution that can efficiently bring together a range of hardware, software and people. Technology solutions range from simplistic to immensely complex and everything in between. We provide a quick summary of typical solutions. We then illustrate the incorporation of our software into middleware layer of the IntelligentRiver™ project. This project provides an ideal platform to build a case for incorporating historically independent monitoring approaches into a single consolidated system. The remote data acquisition challenges faced by the IntelligentRiver™ project mirror those experienced by water resources professionals throughout the state.

Our middleware solution leverages a unique approach among streaming data management software alternatives. We leverage a message oriented middleware called NaradaBrokering to handle the underlying communications of our systems. This message based provides us with a powerful brokered publish/subscribe communications interaction. Building on this foundation, we provide a means for representing data and metadata that leverages existing approaches common to the earth and atmospheric sciences. We detail the relevant aspects of our design in terms of communication interactions and data representation approaches.

The presence of a streaming observation management solution provides a robust platform to develop powerful environmental applications. We present a number of applications that support many of the common functions of an observing system, including archiving, real-time monitoring and quality assurance/checks. We then demonstrate the applicability of this system to large observing systems based on scaling methods and benchmarks

Background.

Environmental observing systems come in many different forms and sizes. The majority of existing solutions act in isolation, addressing a rigid and narrow scope of observation data. The future of observing systems is increasingly dealing with heterogeneous data coming from a wide variety of sources. There are a number of large, continental scale earth observing systems that seek to integrate vast collections of sensed data into consolidated systems. The advantages include greater coordination among researchers and the ability to study the interactions between different types of observations. Furthermore, the economy of scale offers a greater value when dealing with the significant costs associated with developing and maintaining environmental observing systems. With the growing demand comes the need for a secure, high performance software solution capable of supporting geographically and organizationally distributed research teams.

The IntelligentRiver™ project supports a common infrastructure for handling diverse datasets from throughout the state of South Carolina. Currently a wide range of data is being managed including metrological, ground water, surface water quantity and quality. The project includes the incorporation of new instrumentation as well as existing solutions from vendors including Campbell Scientific, Yellow Springs Instruments and Teledyne-Isco. Our approach is intended to be platform independent. Likewise, our system is intended to support a wide range of data applications and repositories.

Traditional methods employed for remote monitoring solutions dictate specific technology choices at every level and are tightly coupled from an architectural point of view. A narrow range of instrumentation is supported, often necessitating proprietary communications protocols. Data is transmitted directly into a purpose built repository, typically a relational database management system. This approach works in static monitoring systems with a fixed number of reporting platforms.

Communications approach plays a key role in the overall design of a streaming data management middleware and is a distinguishing element of our software design. Communications can be characterized by their degree of coupling. Eugster et al. [1] identifies three dimensions of decoupling: space, time and synchronization. Space refers to the anonymity of interactions; participating entities need not be aware of each other. Time specifies that sending and receiving parties need not be present on the system at the same time. Synchronization implies that the program flow of participating parties in not impacted by the communication interaction. Our brokered publish/subscribe communication pattern supports all three decoupling dimensions. Existing data management solutions like RBNB DataTurbine [2] and BRCC Antelope [3] take different approaches with their communications approach and do not support the degree of decoupling possible with a brokered publish/subscribe system.

Methodology.

We use an existing message oriented middleware known as NaradaBrokering [4] to provide our brokered publish/subscribe communications. NaradaBrokering is an open source, general purpose messaging solution developed in Java by the Community Grids Lab at Indiana University. NaradaBrokering can be thought of as an overlay network, supporting the efficient routing and dissemination of messages in a distributed, hierarchically arranged broker network. This approach provides scalability and reliability without a dependence on the availability of a particular broker node. It is well suited for systems with real-time, low-latency communications, including audio/video streaming [5].

The brokered publish/subscribe pattern is comprised of three entities; a publisher, a subscriber and a broker. Communication occurs through the exchange of messages between entities. Messages are routed based on their topic descriptor. Brokers (or a collection of brokers) route messages to subscribers who have registered an interest in a particular topic. This architecture is shown in Figure 1.

Our middleware solution specifies a representational model of data and metadata for transmission over the messaging substrate. By enforcing a structure for data and metadata, we can support powerful and flexible subscribing applications. Data translation capabilities in the middleware can support a range of possible data formats

This middleware provides a basis for building applications that handle data in real-time. Any number of publishing and subscribing entities may act on this data. We have created applications that provide much of the functionality needed in an environmental observing system. For example, real-time monitoring of observation traffic is provided by an AJAX web application that provides a view of currently reporting data sources and their observation data. A novel quality control and assurance application listens to real-time data and provides checks to ensure data fails within acceptable limits, providing instant notification in the event a failure or anomaly occurs. A range of data archiving options are available, including support for NetCDF-3 binary formats, comma separated value text files and a Xenia based RDBMS writer. A flexible publishing application is provided that is able to integrate with a number of existing instrumentation types. Custom development of

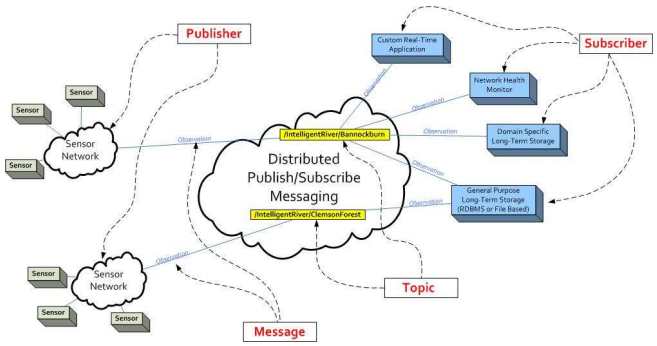

Figure 1 – Publish/Subscribe Architectural View

publishing and subscribing application is possible through the use of a Java-based applications programming interface (API).

Discussion.

The IntelligentRiver™ project currently maintains approximately forty reporting stations resulting in hundreds of observation per minute. We anticipate that this system will grow by orders of magnitude as we bring additional projects online. Additionally, we are looking towards higher resolution instrumentation and remotely sensed data sources. All of this will result in significantly higher demands on our middleware and server infrastructure. Because the foundation of our system is built on a distributed system, we can scale horizontally rather then depending on a single monolithic system to handle increased traffic. Figure 2 shows benchmark results with varying publish rates and message sizes against a single broker node. This is a roundtrip measure from the observation source to the observation consumer, representing the total latency expected for an observation to move from an instrument to its final destination. Test results include 20,000 observations published under rates varying from 10 to 5,000 observations per second. Message sizes vary from 18 bytes to 2,560 bytes. The average transit delay for messages under these high traffic loads results in an average transit delay of 12.31ms.

Conclusion.

A streaming observation data management solution was introduced as a key component of the IntelligentRiver™ project. We characterize our approach and relate it to existing environmental observing systems. A brokered publish/subscribe based system is presented that enables a powerful approach to application development and enables powerful capabilities like real-time quality control/assurance applications. Benchmark results demonstrate the ability of our approach to scale to large systems.

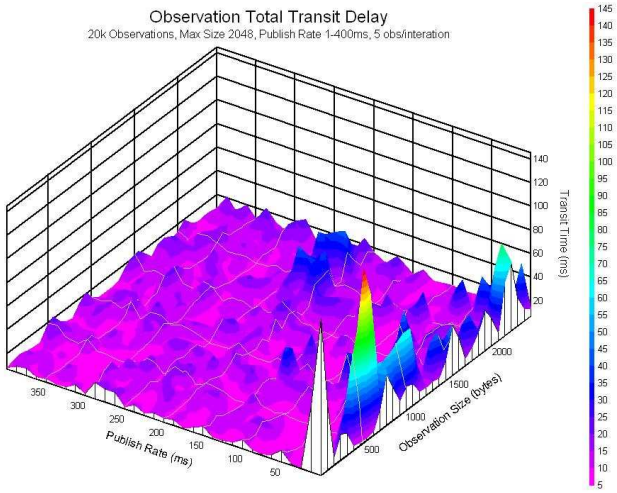

Figure 2 - Observation Transit Delay

Acknowledgements.

Gene Eidson, Jill Gemmill. Clemson Computing and Information Technology (CCIT). The project is sponsored by Clemson Public Service Activities (PSA).

References.

[1] P. T. Eugster, P. A. Felber, R. Guerraoui and A. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys,* vol. 35, pp. 114-131, 2003.

[2] Anonymous (2008). Open Source DataTurbine Initiative. [Online]. Available: http://www.dataturbine.org/

[3] Anonymous (2008). Boulder Real Time Technologies, Antelope System. [Online]. Available: http://www.brtt.com/

[4] Anonymous (2008). The NaradaBrokering Project @ Indiana University. [Online]. Available: http://www.naradabrokering.org/index.html

[5] G. Fox, G. Aydin, H. Bulut, H. Gadgil, S. Pallickara, M. Pierce and W. Wu, "Management of real-time streaming data Grid services: Research Articles," *Concurr. Comput. : Pract. Exper.,* vol. 19, pp. 983-998, 2007.