Emerging Technologies

The Open Source DataTurbine Initiative: Empowering the Scientific Community with Streaming Data Middleware

Tony Fountain (E-mail: tfountain@ucsd.edu), Sameer Tilak (E-mail: stilak@ucsd.edu), Peter Shin (E-mail: pshinn@ucsd.edu), Michael Nekrasov (E-mail:mikrasov@ucsd.edu)

California Institute of Telecommunications and Information Technology, University of California at San Diego, California, USA

Introduction

DataTurbine is a *robust real-time streaming data engine* that lets you quickly stream live data from experiments, labs, web cams and even Java-enabled cell phones. It acts as a "*black box*" to which applications and devices send and receive data. Think of it as express delivery for your data, be it numbers, video, sound or text. For ecological applications, DataTurbine is useful for moving data in near real time from sensors to field stations to data centers. DataTurbine handles time series data over networks with intermittent connectivity. It is vendor-neutral, so it works with sensors and dataloggers from a variety of manufacturers and research labs.

DataTurbine is a buffered middleware product, not simply a publish/subscribe system. In distributed systems, middleware is the software layer that lies between the operating system and the applications running at individual sites (Middleware). DataTurbine can receive data from various sources (sensors, web cams, etc.) and send data to various sinks (visualization interfaces, analysis tools, databases, etc). It has "TiVO" like functionality that lets applications pause and rewind live streaming data.

DataTurbine is **open source** and free. There is also an active developer and user community that continues to evolve the software and assist in applications. The development and publication of DataTurbine is currently supported by the Office of Cyberinfrastructure at the National Science Foundation and by the Gordon and Betty Moore Foundation.

Why Use DataTurbine?

Extendable: It is a free Open Source product with an extensive well-documented API.
Scalable: It uses a hierarchical design that allows your system to grow with the requirements of your application.
Portable: DataTurbine runs on devices ranging from embedded computers and smart phones to multicore servers.
Reliable: Using a Ring Buffered Network Bus, it provides tunable persistent storage at key network nodes to facilitate reliable data transport.
Community Support: There is also an active developer and user community that continues to evolve the software and assist in application development.

Background and motivation

Sensor-based environmental observing systems are essential research platforms for investigating numerous issues that impact basic science, human health, resource management, and economic sustainability. They provide a detailed data record of short- and long-term processes, and in some cases real-time situational awareness and instrument control. However, designing, building, and operating an observing system of even modest scale presents technological and organizational challenges. A solid software foundation can mitigate some of these challenges. DataTurbine is a mature and robust open-source software product for managing real-time sensor streams. It addresses a core set of requirements that are common across a number of applications in ecology and environmental engineering. It was specifically designed to support streaming-data applications that integrate heterogeneous sensors and need reliable sensor data transport over unreliable networks. At present, OSDT is the only open-source streaming data middleware system available to NSF science and engineering communities.

Distributed sensor-based applications range from passive observing systems to dynamic closed-loop feedback systems involving feedback and control. Some systems only need to collect sensor data at a fixed rate; other systems require more sophisticated interactions including event detection and adaptive sampling. They all face numerous software challenges due to requirements for *heterogeneous instrumentation* and *complex real-time processing* over *networks with intermittent connectivity*. Systems must incorporate instruments from across the spectrum of complexity, from accelerometers, to meteorological stations, to acoustic Doppler current profilers, to streaming video cameras. They require the integration of numerous hardware components (e.g., servers, routers, and radios) and software modules (e.g., QA/QC, modeling, visualization, and data management). These systems must operate under a variety of networking conditions, including wired and wireless, persistent and intermittent. And they have stringent requirements on data timeliness and integrity. Data must be reliably captured, moved to acquisition and control points, and processed efficiently so that decisions can be made and control actions can be executed within acceptable temporal constraints. Keeping up with the rate of data generation, moving data reliably across networks with intermittent connectivity, storing and processing it all, are key requirements for these systems. The DataTurbine middleware was designed and developed to provide streaming data software support for building and managing these types of applications.

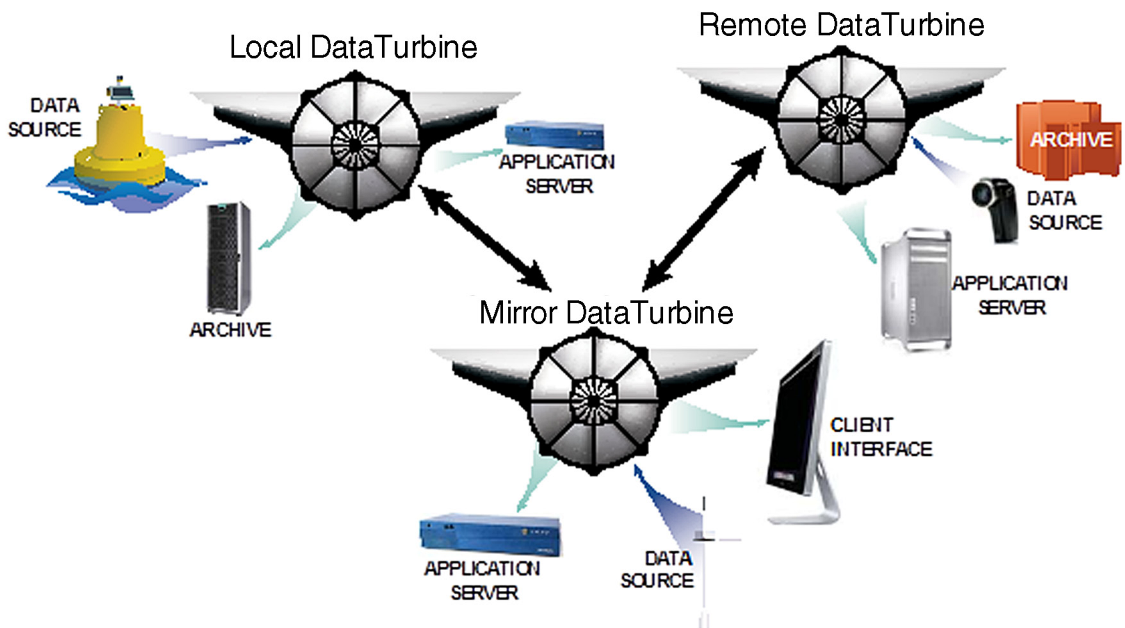

Fig. 1. DataTurbine middleware provides the cyberinfrastructure that integrates the disparate elements of complex distributed real-time applications.

How DataTurbine works

DataTurbine supports the development and operations of sensor-based systems by consolidating complex data stream management tasks into a uniform framework that is modular, scalable, and robust. DataTurbine provides (1) a programming abstraction over heterogeneous devices, and (2) a suite of integrated network services for managing streaming data. DataTurbine makes disparate devices look similar, e.g., data streams from instruments as diverse as accelerometers and video cameras are easily integrated and managed by a common Application Programming Interface (API). From the perspective of distributed systems, the DataTurbine middleware is a "black box" to which applications and devices send and receive data (Fig 1). DataTurbine handles important data management operations between data sources and sinks, including reliable transport, routing, and scheduling. DataTurbine accomplishes this through the innovative use of flexible network bus objects combined with memory- and file-based ring buffers. Network bus objects perform data stream multiplexing and routing. They organize time series data into channels and frames for efficient processing. The ring buffers provide tunable persistent storage at key network nodes to facilitate reliable data transport. Ring buffers also connect directly to client applications to provide TiVo-like services including data stream subscription, capture, rewind, and replay. This presents client applications with a uniform interface to real-time and historical (playback) data. DataTurbine also provides bookkeeping services and reconnect logic to enable a level of disruption-tolerant networking. When a system interruption occurs, DataTurbine captures the state of data transmissions. When the system recovers, DataTurbine reconnects and starts sending data from that point. This feature is useful in distributed applications where network stability is an issue. DataTurbine is also portable and scalable. Our tests demonstrate that DataTurbine runs efficiently on platforms from smart phones to supercomputers (Tilak et al. 2007). It has been successfully deployed in science and engineering applications funded by NSF, NASA, NOAA, DoD, and private industry (Fountain et al. 2009; NASA-NAMMA, PME).

System architecture for environmental observing systems

In designing an environmental observing system, it is useful to consider a generic system architecture. This architecture provides an abstraction over specific types of sensors, computers, and networks so that general issues of design and performance can be considered and evaluated. A typical environmental observing system consists of two major components (1) the field-deployed sensor networks (aka embedded cyberinfrastructure (CI)), and (2) a Data Center (sometimes as simple as a field station computer). The embedded CI is typically a heterogeneous multi-hop sensor network that consists of three tiers: (1) Sensors, (2) Cluster Heads (CH), and (3) Gateway nodes (GW). Fig. 1 depicts the key entities and relationships that define this architecture. Briefly, sensors are connected to Cluster Heads. A Cluster Head (CH) is a lightweight compute platform that resides close to the deployed instrumentation. In some systems, a traditional datalogger serves as the cluster head. Cluster heads are then organized into local networks and in turn connect to a Gateway node using one or more of the following networking technologies: 802.11 or cellular network, satellite radios, or even Ethernet. In simple systems the cluster head is also the gateway node and the network is a simple point-to-point single hope, e.g., from buoy or tower to field station. The Gateway node (GW) in turn connects to a Data Center over the Internet. Again, a Data Center can be as simple as a field station server to a remote compute cloud such as Amazon. A GW node is essentially an interface between the embedded CI and the Data Center. Client applications connect to the Data Center to access the data.

The OSDT android controller

Although many applications use DataTurbine simply to fetch data from conventional dataloggers and then move it to a data server, there is increasing interest in moving DataTurbine into the field. Scientists would like to (1) integrate heterogeneous sensors that are not easily integrated with conventional dataloggers, and (2) perform additional processing close to the sensors, in particular event detection and control. In general there is a need for more complex algorithms to be running close to the sensors. To address these requirements, we developed the OSDT Android Controller. The OSDT Android Controller runs on any Android platform, e.g., smart phone or tablet, and provides the full DataTurbine functionality along with a suite of software modules for configuration, scheduling, and control. As part of the project, we ported the OSDT middleware to the Android platform and developed new software for configuring and managing real-time embedded applications. Although it was initially targeted to smart buoy controller applications, the core is a generic embedded/mobile streaming data system. By

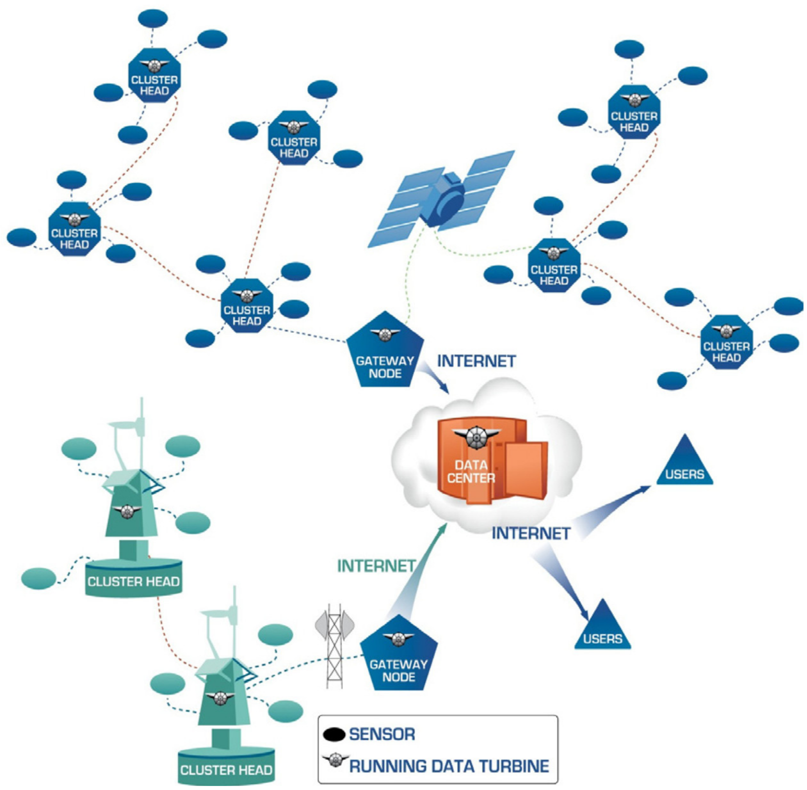

Fig. 2. System architecture for environmental observing systems.

employing standard Android operating system, the developed software is readily available on a broad range of devices including smartphones, tablets, and netbooks. In essence, choice of Android platform allows us to leverage the tremendous engineering investment made in producing what has become commodity-embedded systems.

The OSDT-Android controller manages sensor interfaces, data acquisition, on-board processing, and data transmission over multiple types of radios, including Iridium satellite, cellular, Bluetooth, and long-distance wireless. When combined with a Droid cell phone or tablet, the OSDT Android controller becomes a robust sensor pod that can be configured to serve as a cluster head or gateway node in complex sensor-based systems. The Sensor Pod software architecture is built around the OSDT middleware. The OSDT acts as a message bus between software components and provides a suite of services for data management, buffering, routing, synchronization, and monitoring. The software is developed in Java. It is easily reprogrammable and it includes software for scheduling sensor operations and communications. It is designed to support on-platform event detection and real-time control. It can be readily updated to incorporate new software modules, and dynamically reconfigured to schedule these modules to control sensor operations and communications. To inform the technical design and provide valuable field experience, we are collaborating with several groups to test the OSDT Sensor Pod. This includes researchers at NTL LTER site (Wisconsin, United States), the MCR LTER site (Moorea, French Polynesia), Palmyra Atoll (Central Pacific), and the Taiwan Forest Research Institute (Taipei, Taiwan).

Example applications

In this section we describe some of the key applications in which we are involved with the OSDT Initiative. In each application we provide an overview of the research issue and the role of OSDT in the system.

F.1 Ocean Acidification at Coral Reefs. While coral reefs have undergone unprecedented changes in community structure in the past 50 years, they now may be exposed to their gravest threat since the Triassic. This threat is increasing atmospheric CO_2, which equilibrates with seawater and causes ocean acidification (OA). Key to understanding this process is measurements by OA-related sensors including pH, PCO_2, temperature, and pressure. The deployment of OA-related sensors at MCR and Palmyra Atoll and the collection of measurements from these sensors will provide valuable insights into OA and lay the foundation for larger, more rigorous OA studies in the future. The ability to measure these phenomena in situ and in real time represents a significant advance in our ability to monitor key environmental processes. The OSDT Android Sensor Pod will provide a critical building block for enabling new science experiments on OA, and the results will be applicable to other coral reef researchers.

F.2 Lake Mixing to Eradicate Invasive Species. Rainbow smelt, a cool-water fish species, once established, lead to the demise of native fish species, such as yellow perch and cisco (or lake herring), due to competition for food resources or direct predation. In addition, walleye are unable to reproduce successfully in lakes where smelt are present, leading to reduced populations of this important recreational game fish. A new approach to eliminating rainbow smelt takes advantage of the differences in thermal tolerances between smelt, yellow perch, and walleye. Smelt require cooler water temperatures than either yellow perch or walleye, and consequently live in the deeper, cooler parts of lakes. By experimentally mixing a lake during midsummer we can warm the deeper waters of a lake, eliminating the cool-water habitat required by the smelt. Key to the success of this experiment is collecting associated measures of lake processes, including dissolved oxygen, chlorophyll, and temperature at multiple depths. The OSDT Android Sensor Pod will be deployed as the sensor and communications manager on a suite of buoys for the lake mixing experiment. The scientific results will provide important insights into lake processes and serve as a model for future sensor deployments in The Global Lake Ecological Observatory Network (GLEON).

F.3 Free Ocean Carbon Enrichment Experiment. Monterey Bay Aquarium Research Institute (MBARI: http://www.mbari.org/) scientists and engineers have designed FOCE (Free Ocean Carbon Enrichment) technology to enable the study of the impact of this pH change on ocean biogeochemistry and ecology. The FOCE technology concept enables small-scale, in situ CO_2 enrichment experiments to be carried out, in a manner analogous to the terrestrial Free Air CO_2 Enrichment (FACE) experiments (Long et al. 2006). FOCE is used to control the pH within a small volume of seawater that exchanges freely with the surrounding region in the ocean. The technology uses feedback from pH sensors and other instruments to inject CO_2, creating ocean acidification conditions (reduced pH) in a small area, corresponding to future levels of greenhouse gases, while maintaining other environmental parameters.

In this way, FOCE is like an ocean acidification time machine, allowing researchers to peer into the ocean's future to see the effect on natural ecosystems such as coral reefs, cold-water corals, and other sensitive benthic habitats. DataTurbine provides asynchronous communication links between distributed components in FOCE setup, and is particularly well suited to streaming instrument data.

F.4 Structural Health Monitoring. Advanced Hazards Mitigation Laboratory at University of Connecticut. Structural health monitoring can provide an unbiased vibration-based assessment of the structural infrastructure in a timely and efficient manner (ADVANCED HAZARDS MITIGATION LABORATORY). This is critical in our society, faced with an aging infrastructure and limited resources for maintenance and repair. Bridge monitoring in Connecticut is a combined effort between the University of Connecticut and Connecticut Department of Transportation. This program of short- and long-term monitoring currently has a network of six bridges with long-term monitoring systems. DataTurbine meets a need to provide fully automated continuous monitoring from remote locations, and can be used to effectively convey the results of bridge monitoring to the end user. DataTurbine was deployed to stream data from accelerometers, strain gages, and video cameras installed on two of the highway bridges in Connecticut.

Related technologies

The OSDT middleware is unique in its combination of features and its focus on distributed streaming data sensor-based systems. It was designed to address requirements from these types of applications. It is most often compared to (1) messaging systems (Liu and Plale 2003), (2) database systems (MySQL, PostgreSQL, Xstream (Girod et al. 2008), (3) datalogger software (LoggerNet), (4) complex event stream processing engines (ESPER, StreamBase, StreamInsight), and (5) proprietary real-time control software packages, e.g., LabVIEW. Messaging systems such as MSMQ, Websphere MQ, NaradaBrokering (Pallickara and Fox 2003), Enterprise messaging systems (EMS), Enterprise Service Bus (Chappell 2004), Java Message Service (JMS), and various publish–subscribe systems (Liu and Plale 2003) provide support for guaranteed messaging. It can be argued that they provide a reasonable foundation for building streaming data science applications. However, in their current form they fail on many science and engineering requirements. In general, they were not developed with sensors and science applications in mind, e.g., requirements for integrating heterogeneous instruments, performing in-network processing, and managing time series data. OSDT was designed from the beginning to address these requirements. For example, OSDT can simultaneously support both pull (application requests data from the middleware) and push (middleware uses callback functions to push data to applications) modes over intermittent networks with guaranteed data delivery. It also provides tunable persistent storage at network nodes to support complex application demands. These features are at the core of OSDT and were motivated by real-world applications where data needs to be pulled from various devices and pushed to other devices during windows of network availability. The modularity of OSDT makes it straightforward to design complex applications that leverage the strengths of these other tools. Since it is an open-source software product, it avoids the limitations of proprietary software, including their lack of extensibility and the threat of commercial vendor lock-in. At present, OSDT is the only open-source streaming data middleware system available to NSF science and engineering communities.

Future work

We surveyed the current state of the OSDT Initiative and identified two technological areas that are important for moving forward: (1) software extensions that allow DataTurbine applications to run in a cloud-computing environment (Chappell 2008, Nurmi et al. 2008). Cloud computing (i.e., the integration of virtualization, infrastructure as a service (IaaS), software as a service (SaaS), and utility computing) provides a scalable and cost-efficient computing paradigm. Cloud computing addresses two requirements that are common across sensor networks and observing systems. The first is support for scalable services (both infrastructure services and application services). The second is application and project sustainability. Cloud computing enables application and project sustainability via drastic cutback in infrastructure startup, expansion, and maintenance costs as well as typical personnel costs associated with system operations. Data from field-deployed sensor networks can be streamed to a cloud platform for processing and sharing among collaborators. (2) software interfaces that are compatible with the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) standards (Botts et al. 2007, OGC, OGC-SWE). The OGC-SWE framework provides service definitions, interaction models, and encoding standards for designing, building, and operating sensor-based systems. Taken together, the OGC-SWE framework and the cloud-computing paradigm comprise a principled approach to building, deploying, and operating scalable sensor-based streaming-data applications.

In addition, we are planning to build a Full-Featured OSDT-Matlab Interface. Matlab is a popular programming and modeling environment for many scientific and engineering communities. There currently exists a basic OSDT interface to Matlab that supports limited interactions between the two software tools. However, the limited functionality and documentation, and the inefficiencies of a cross-tool programming model (Java and Matlab) are constraints on productivity for scientists, systems developers, information managers, and students. The current OSDT-Matlab interface exposes a Java-style syntax. Java and Matlab differ significantly in terms of syntax and style. To that end, we plan to develop a new "full-featured" OSDT Matlab interface that includes all the functionality provided by the Java-based interface. The new interface will create a Matlab-style interface to OSDT that is efficient and easy to use.

Summary

The Open Source DataTurbine Initiative is an international community of scientists and engineers sharing a common interest in real-time streaming data middleware and applications. The technology base of the OSDT Initiative is the DataTurbine open-source middleware. The DataTurbine middleware satisfies a core set of infrastructure requirements that are common in sensor-based systems, including reliable data transport, a framework for integrating heterogeneous instruments, and a comprehensive suite of services for data management, routing, synchronization, monitoring, and visualization. It has been demonstrated to run efficiently on platforms from cell phones to supercomputers. It has been successfully deployed in many real-world environmental monitoring applications including coral reef monitoring and lake monitoring. We plan to develop extensions that will allow DataTurbine applications to run on a cloud platform to improve the system scalability and project sustainability. We are also planning to develop OGC-SWE compliant interfaces to DataTurbine. These standards-based interfaces will promote interoperability of environmental observing systems.

References

AHML (Advanced Hazards Mitigation Laboratory) http://www.engr.uconn.edu/~richard/lab/projects.html

Apache2.0 (Apache License Version 2.0) http://www.opensource.org/licenses/apache2.0.php

Barseghian, D., I. Altintas, M. B. Jones, D. Crawl, N. Potter, J. Gallagher, P. Cornillon, M. Schildhauer. E. Borer, E. Seabloom, and P. Hosseini. 2010. Workflows and extensions to the Kepler scientific workflow system to support environmental sensor data access and analysis. Ecological Informatics 5:42–50.

Benson, B. J., B. J. Bond, M. P. Hamilton, R. K. Monson, and R. Han. 2010. Perspectives on next-generation technology for environmental sensor networks. Frontiers in Ecology and the Environment 8:193–200.

Benson, B. J., L. Winslow, P. W. Arzberger, C. C. Carey, T. Fountain, P. C. Hanson, T. K. Kratz, and S. Tilak. 2008. Meeting the challenges of an international grassroots organization of sites deploying sensor networks: the Global Lake Ecological Observatory Network (GLEON). Proceedings of the Environmental Information Management Conference.

Botts, M., G. Percivall, C. Reed, and J. Davidson. 2007. OGC Sensor Web Enablement: overview and high level architecture. OGC White Paper.

Chappell, D. A short introduction to cloud platforms: an enterprise-oriented view. http://www.davidchappell.com/CloudPlatforms--Chappell.pdf

Chappell, D. A. 2004. Enterprise service bus. O'Reilly Media. ISBN 13: 9780596006754.

CREON: The Coral Reef Environmental Observatory Network. http://www.coralreefeon.org/

Dyke, S. J., R. E. Christenson, and S. Courtner. 2010. Tele-operation shake table experiments in earthquake engineering for undergraduate education. Proceedings of the EERI Conference, Toronto, Canada, June, 2010.

EMS (Enterprise messaging systems) http://en.wikipedia.org/wiki/Enterprise_Messaging_System

Esper. Event Processing for Java. http://www.espertech.com/products/esper.php

Fountain, T., S. Tilak, P. Hubbard, P. Shin, and L. Freudinger. 2009. The Open Source DataTurbine Initiative: streaming data middleware for environmental observing systems. The 33rd International Symposium on Remote Sensing of Environment, Stresa, Italy, 4–8 May 2009. http://www.dataturbine.org/biblio

Fountain, T., S. Tilak, P. Shin, S. Holbrook, R. J. Schmitt, A. Brooks, L. Washburn, and D. Salazar. 2009. Digital Moorea Cyberinfrastructure for Coral Reef Monitoring. International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), December, 2009.

Girod, L., Y. Mei, R. Newton, S. Rost, A. Thiagarajan, H. Balakrishnan, and S. Madden. 2008. XStream: a signal-oriented data stream management system. Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE), 2008.

GLEON: The Global Lake Ecological Observatory Network. http://www.gleon.org/

Jaroensutasinee, M., K. Jaroensutasinee, T. Fountain, M. Nekrasov, S. Chumkiew, P. Noonsang, U. Kuhapong, and S. Bainbridge. 2011. Coral sensor network at Racha Island, Thailand. Environmental Information Management.

JMS (Java Message Service) JSR 914: API. http://jcp.org/en/jsr/detail?id=914

LabVIEW. National Instruments: The LabVIEW Environment. http://www.ni.com/labview/

Liu, Y., and B. Plale. 2003. Survey of publish–subscribe event systems. Technical Report TR574, Indiana University, May, 2003.

LoggerNet. Campbell Scientific: LoggerNet datalogger support software. http://www.campbellsci.com/loggernet3x

Long, S. P., et al. 2006. Food for thought: lower than expected crop yield stimulation with rising CO_2 concentrations. Science 312:1918.

Middleware. What is Middleware. http://middleware.objectweb.org/

MSMQ. Microsoft Message Queuing MSMQ technology. http://www.microsoft.com/windowsserver2003/technologies/msmq/default.mspx

MQ. IBM WebSphere MQ. http://www.ibm.com/software/mqseries/

MySQL. http://www.mysql.com/

NASA-NAMMA. NAMMA: The NASA African monsoon multidisciplinary analyses program. http://namma.msfc.nasa.gov/

Nurmi, D., R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. 2008. The Eucalyptus open-source cloud-computing system. Proceedings of Cloud Computing and Its Applications, Chicago, Illinois, USA. October, 2008.

OGC. Open Geospatial Consortium. http://www.opengeospatial.org/

OGC-SWE. Open Geospatial Consortium–Sensor Web Enablement Initiative. http://www.opengeospatial.org/projects/groups/sensorweb

O'Reilly, T., K. Headley, R. Herlien, K. A Salamy, S. Tilak, D. Edgington, T. Fountain, P. Brewer, and W. Kirkwood,. A sensor network platform to study impact of ocean acidification in deep-water environments. In the demonstration session at the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS) 2010.

Pallickara, S., and G. Fox. 2003. NaradaBrokering: a middleware framework and architecture for enabling durable peer-to-peer grids. Proceedings of ACM/IFIP/USENIX International Middleware Conference on Middleware. 2003:41–61.

PME. Precision Measurement Engineering, Inc. LakeESP (Environmental Sensing Platform. http://www.pme.com/HTML Docs/LakeESP.html

PostgreSQL: Object-relational database system. http://www.postgresql.org/

Strandell, E., S. Tilak, H. M. Chou, Y. T. Wang, F. P. Lin, P. Arzberger, and T. Fountain. 2007. Data management at Kenting's underwater ecological observatory. *In* Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), December, 2007.

StreamBase complex event processing (CEP) platform. http://www.streambase.com/

StreamInsight. Microsoft StreamInsight platform. http://msdn.microsoft.com/en-us/library/ee362541.aspx

Tilak, S., P. Arzberger, D. Balsiger, B. Benson, R. Bhalerao, K. Chiu, T. Fountain, D. Hamilton, P. Hanson, T. Kratz, F. P. Lin, T. Meinke, and L. Winslow. Conceptual challenges and practical issues in building the Global Lake Ecological Observatory Network. *In* Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), December, 2007.

Tilak, S., P. Hubbard, M. Miller, and T. Fountain. 2007. The Ring Buffer Network Bus (RBNB) DataTurbine streaming data middleware for environmental observing systems. Proceedings of the Third IEEE International Conference on E-Science and Grid Computing (e-Science), 2007.