Century: Automated Aspects of Patient Care♦

Marion Blount*, John Davis*, Maria Ebling*, Ji Hyun Kim**, Kyun Hyun Kim**, KangYoon Lee**, Archan Misra*, SeHun Park**, Daby Sow*, Young Ju Tak**, Min Wang*, Karen Witting*
**IBM T.J. Watson Research Center and **IBM Korea Ubiquitous Computing Lab*
mlblount@us.ibm.com, davisjs@us.ibm.com, ebling@us.ibm.com, jihkim@kr.ibm.com, andrew.kim@us.ibm.com, keylee@kr.ibm.com, archan@us.ibm.com, sehpark@kr.ibm.com, sowdaby@us.ibm.com, yjtak@kr.ibm.com, min@us.ibm.com, witting@us.ibm.com

Abstract

Remote health monitoring affords the possibility of improving the quality of health care by enabling relatively inexpensive out-patient care. However, remote health monitoring raises new a problem: the potential for data explosion in health care systems. To address this problem, the remote health monitoring systems must be integrated with analysis tools that provide automated trend analysis and event detection in real time. In this paper, we propose an overview of Century, an extensible framework for analysis of large numbers of remote sensor-based medical data streams.

1. Introduction

Recently, astronomical health care costs in conjunction with increasingly affordable networked sensors have resulted in a new driving force behind the adoption of remote health monitoring. Health care expenditures in the US have exceeded $1.9 trillion [2] and a 2006 Organization for Economic Co-operation and Development (OECD) report projects that annual health care expenditures for the 30 OECD member nations will nearly double between 2005 and 2050, due in large part to aging populations. Fortunately, these cost increases may be countered by an army of inexpensive, networked, wearable sensors that offer the opportunity of using remote health monitoring, not just for secluded patients, but as a way to reduce health care costs system-wide. Remote health monitoring affords the possibility of improving care by enabling relatively inexpensive out-patient care for patients with chronic conditions and by efficiently managing the response to acute medical conditions.

While remote health monitoring offers a solution to the health care problems cited above, it also has the potential of introducing a new problem: *data explosion*. Consider the case of diabetes, a disease that currently costs $132 billion annually in the US alone and is projected by the World Health Organization to increase from 177 million patients in 2000 to over 300 million patients in 2025 [1]. Blood glucose monitoring is an important regiment that diabetes patients must undergo daily; typically four glucose readings are taken per day, with the patient writing down the values in a log book. Alternatively, the use of commercially available networked glucometers can automatically and accurately replace this error prone manual glucose reading process. The wide spread usage of these devices by the 13 million diabetes patients in the US served by 2,728 office-based endocrinologists (according to figures recorded in 2002 [3]) would generate a total of 52,000,000 glucose readings daily; almost 20,000 readings per day for each endocrinologist! Clearly, no physician can handle the quantity of glucose readings calculated above (even if delivered in paper form).

This data explosion is exacerbated by the emergence of sensors producing data at high rates, such as electrocardiograms (ECG) and gait monitors. Clearly, remote health monitoring systems must be designed to do more than simply collect and store data.

We claim that such systems should be integrated with analysis tools that provide automated trend analysis and event detection on the care provider's behalf. We envision an ecosystem of hardware OEMs, independent software vendors, service providers, hospital IT administrators, government agencies, academics and physicians providing and/or selecting the sensors, algorithms and analysis components that process patient data. Such systems integrated with a large, dynamic community of players motivate several important needs. First, we need a system architecture that scales to accommodate high event throughput and handles distributed system components. Second, we need a well defined programming model that enables the design, integration and deployment of new data sources and analysis components. Third, we need a way to compactly store and efficiently interrogate the provenance

♦ This work was supported by the IT R&D program of MIC/IITA under the project/grant/funding number 2006-S-602-01 (Development of Stream-based Distributed Interoperable Health care Infrastructure Supporting Provenance and QoE).

13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications(RTCSA 2007)
0-7695-2975-5/07 $25.00 © 2007 IEEE

of events generated by such sensors and analysis components, for applications including treatment validation, anomaly analysis and data replay.

In this paper we describe an extensible framework (referred to as *Century*) that enables real-time analysis of remote medical sensor data streams and supports features such as stream data provenance.

2. Related Work

Several research groups have proposed systems allowing modern medical practitioners to extend their services to patient homes. For example, Blount et al [9] have proposed systems for delivering health data from body-worn sensors through a networked hub. Our work differs from these approaches in several ways. Our approach is open to a large spectrum of data sources ranging from very low rate data sources like blood pressure cuffs to high rate data sources like ECGs and EEGs. In addition, Century provides an extensible framework where application developers can design monitoring and analysis applications. Finally, support for data provenance is embedded into the core infrastructure, allowing stakeholders to validate and track the origin of events processed or generated by the system.

Stream processing is an active research area in academia and plays a key role in the development of Century. Stream database systems are an emerging class of data management systems designed to specifically handle streaming data. The focus of these systems is on the ability to make long standing queries on data streams. There are several projects such as Aurora [12] and Borealis [13] that have proposed stream database systems with the goal of allowing applications to manipulate and query stream data from a database-centric perspective. They typically apply fixed sets of analysis operators on the streams. Queries for data still follow the models used in traditional relational database systems. The SPC middleware [10] extends this concept by allowing the stream analysis logic to include not only relational operators, but also user-defined operators. All this body of work is complementary to Century. Century may be viewed as a specialization of such stream processing systems to the health care domain, with optimized support for health care domain-specific requirements, such as stream persistency, very large sets of sensor streams and stream provenance.

3. Requirements

Century addresses several specific requirements imposed by the remote health monitoring ecosystem. This ecosystem consists of a physical infrastructure and participants that have a stake in the data collected on patients.

3.1 Infrastructure Requirements

Consistent with much of the sensor network literature, we model the physical components of a remote health monitoring system with a three tiered architecture [6]: a sensor tier, a data hub tier and a server tier.

The *sensor tier* consists of a wide spectrum of sensors, ranging from biometric sensors collecting physiological data about a person to environmental sensors collecting information about the environment in which the person lives. Coping with such a large amount of heterogeneity dictates the design of an architecture that is open and extensible to unanticipated data sources. Furthermore, this architecture must cope with very large numbers of sensor data sources. This number is directly proportional to the number of patients in the system, and creates interesting device management and scalability issues.

The second tier in the system is the *data hub* tier. Many sensors do not have wide area networking capabilities and are limited to local communications capabilities (e.g., BlueTooth, ZigBee, 802.11). The hub serves as a store and forward mechanism that either resides on a mobile device (e.g., a cellular phone) or a static machine (e.g., a desktop computer). It is worth noting, that the data hub may be used as a gateway between the sensors and server in part because the server simply doesn't need to see all of the collected data. For example, the data hub may be configured to only forward anomalous data that signals emergency medical events. In such cases, analysis capabilities are needed at the data hub. Moreover, the distributed operation of the analysis components must also factor in the device's energy and bandwidth constraints.

The *server tier* is where the sensor data is stored and the majority of data processing occurs. In most cases medical data is never deleted, so the server must be capable of managing storage, backup, replication, and recovery of data. Complex queries may be processed in the server tier to access the desired data. Moreover, stream processing algorithms may be applied to discover medically significant events in this ocean of streaming data (on individual users or across different sets of user populations) received at the server tier. The type of analysis performed is dependent on the condition of the patients and the intent from the stakeholders. Hence, there is a need for an open platform where arbitrary analysis algorithms are instantiated.

3.2 Stakeholders Requirements

There are three broad groups of people who will interpret the data collected via remote health monitoring. One group consists of medical professionals, including doctors and nurses, who use the analysis of remote monitoring data as input in the patient's treatment. These persons are *unlikely* to invest the time to learn query languages or to input detailed query parameters in order to get analysis results. They are also *unlikely* to wait multiple minutes for an analysis routine to run. They are *likely* to want the most

13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications(RTCSA 2007)
0-7695-2975-5/07 $25.00 © 2007 IEEE

precise results that require the most sophisticated analysis. Finally, when receiving analysis results they subscribed to (e.g., alerts for medically significant events), they are *likely* to issue provenance queries to track the origin of these results and attach such provenance data to appropriate medical reports (such as electronic health records).

The next group who will use the remote health data consists of medical researchers. Medical researchers will mostly perform off-line analysis of large amounts of remote monitoring gathered from a statistically significant population. They will investigate various algorithms for determining medically significant events, perform clinical trial analysis and apply data mining to find interesting patterns and correlations. This group will invest the time to master and use any tools provided to assist in their analysis. They are also *likely* to issue provenance queries for *a)* the recipe used for the generation of scientific results (so that they reproduce or modify it) or *b)* the relevant past history of raw or derived medical events (so that they may replay such data with newer, more promising analysis tools).

The last group consists of the patient and the patient's family, friends, and caregiver network. Rather than present views of the raw data and detailed medical analysis, high level summaries and general trend information will be most useful and should be made available to this group.

4. Century Server Architecture

The requirements outlined in Section 3 are addressed by the Century infrastructure described in , with the exception of the ability to distribute the event analysis capabilities to the data hub (which we plan to address in the future). The Century server is at the heart of our approach to address the data explosion problem introduced in Section 1. It provides an environment where events that are unimportant to stakeholders can be filtered out. It consists of four components: the stream processing middleware, the patient registration and management subsystem, the event store and the provenance subsystem.

4.2.1 The Stream Processing Middleware

The role of the stream processing middleware is to orchestrate the analysis of high volumes of data streams generated by a large user population. We use the IBM System S Stream Processing Core (SPC) [8] as a base for this middleware. SPC is a scalable distributed middleware for generic stream processing, built to support applications analyzing large volumes of streaming data. The SPC programming model revolves around the concept of a *Processing Element* (PE). PEs are building blocks used to create applications that uses SPC services. PEs communicate with each other through ports. Following a publish-subscribe paradigm, input ports are used by PEs to read data published by other PEs while output ports are used to write data to be consumed by other PEs. The data exchanged between PEs are referred to as Stream Data Objects (SDOs) and are strongly typed according to an extensible type system provided by SPC. For example, a PE extracting specific features from ECG streams (e.g., the QRS interval) would have one of its input ports subscribe to SDOs of type ECG. The Century data model is essentially an extension of the SPC type system. Currently, we support blood pressure, glucose, weight and ECG events. In the future, we anticipate a significant role of health care standard bodies in the specification of this data model.

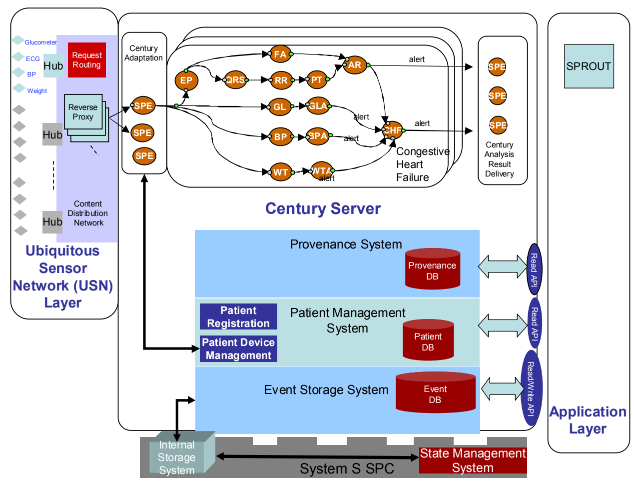

Figure 1: End to end overview of Century

There are three major classes of PEs supported in SPC: source PEs, transform PEs and sink PEs. Source PEs form the Century Adaptation component. They leverage the Century registration mechanism to admit legitimate requests from users registered in the system. They also serve as data adapters listening for incoming requests from data hubs. They adapt these requests and produce SDOs at their output. For scalable operation, each SDO generated by a source PE handling continuous medical streams (e.g., ECG readings) will typically contain a "chunk" of raw data samples (e.g., all ECG readings aggregated over disjoint 1 minute intervals). Transform PEs are atomic elements used for analysis. They encapsulate the logic of the analysis algorithms that need to be applied to the streams entering the server tier. Sink PEs do not have output ports. As their name indicates they reside at the end of the analysis flow. Their role consists of delivering analysis results to the Century Delivery System. This system interfaces with external applications such as external medical data stores or stakeholders client applications interested in being notified when medically significant events have been detected. For example, a nurse might subscribe to SMS notifications sent when the condition of a patient with arrhythmia is deteriorating. In this case a Century sink PE might forward arrhythmia alerts the delivery system who knows how to dispatch the message effectively.

13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications(RTCSA 2007)
0-7695-2975-5/07 $25.00 © 2007 IEEE

In many cases, PEs are stateful, implying that the precise processing logic applied to an incoming SDO may depend on the temporal history of past SDOs or some other external "context". For instance, a PE performing trend analysis of weight on a per patient basis needs to retrieve patient state information as events flow through it. Century extends SPC and provides developers with a State Management System where state information can be persisted and queried. Behind the State Management System resides a storage area maintained by SPC.

4.2.2 Patient Registration and Management Subsystem

Users are registered in Century at the server tier, via the Patient Registration and Management subsystem. The role of this component is twofold. First, it allows Century to manage patient records for all subscribed users. It leverages a patient registration module that is front-ended with a web-based application consisting of a set of JSPs through which administrators can associate a Century user ID to patients and enter static information about patients such as demographics. In addition, portions of patient health records might be downloaded and stored in the Patient Database. We envision the content of this database to be maintained exclusively by medical professionals. As a result, Century software components have read-only access to it. In the future, we plan on extending the functions of this module to allow it to connect itself to electronic medical records in a standardized way.

The second role of this subsystem is to track all the devices registered in Century. To this end, the registration process also captures the MAC addresses of data hubs associated with users, as well as detailed information about sensor kits that are transmitting events on their behalf. A patient device management module resides in the patient registration and management system to perform this tracking.

4.2.3 The Event Store

Century storage is a combination of the stream storage system in SPC and the Event Store. The SPC storage system is optimized for the reception and storage of high speed, large volume streams. SPC enables a stream associated with a given PE to be selectively persisted in the SPC store. Any data in the SPC store can be streamed to PE's based on dynamic subscriptions made by the PE. One assumption made by designers of the SPC store is that the data is not retained for long periods of time (say years); rather, it is retained for a relatively short time period (e.g., a few hours or days) during which it may be potentially needed by another PE. The SPC store scales from a single disk to large SAN-based systems. Traditional file systems and databases are not sufficient to meet the requirements of the SPC store. At any point in time, the SPC contains a number of store areas that have specific policies like retention policy, data value policy, replacement policy, and others. The system uses autonomic control to manage the SPC store based on the policies in effect.

The Event Store provides long term persistence of data and remote access to the data in a traditional database called the Event DB. The key technical challenges are interfacing the SPC store to the Event DB and maintaining the Century data model for remote assessors. Our approach is to use the SPC store as a temporary buffer. The persisted copies of streams in the SPC store are tagged with a special "CenturyPersistedStream" type defined in our type system. The event store shapes the traffic directed towards Event DB to ensure the stability of the system. To perform this task, the event store instantiates a set of special PEs that subscribe to this "CenturyPersistedStream" type and then insert the corresponding streams into the database, at rates that can be sustained by the database. It also provides a remote data access API based on the Century type system, for external application to access persisted streams and analysis results. A design has been completed and the prototype is being implemented.

4.3 The Provenance Subsystem

A variety of legal, legislative and medical requirements will make it mandatory for automated health care systems to support provenance-related functions such as "dependency analysis" or "data replay" needed by stakeholders. The Stream Provenance subsystem thus needs to automatically collect and store metadata whenever analysis components (internally represented as PE graphs) are instantiated or destroyed. Such metadata is required to support queries such as a) "What low-level sensor data contributed to (or did not) to this automated alert", or b) "which stream processing components (PEs) did this alert depend on?" that may be issued by medical professionals for audit or compliance purposes in the future. Prior work on data provenance has argued for one of two approaches. *Annotation-based provenance* appends metadata to the underlying data (e.g., PASS [22] tags each individual data sample with metadata describing provenance details). The alternative *process-based provenance* approach stores only connection-level information among processes (e.g., PASOA [21] stores as provenance the descriptors of the Web services that consume and produce data in a given workflow). Both of these approaches were developed for relatively low data rate, memoryless transactional applications. The high volume of medical events generated by the data streams in Century invalidates both of these approaches.

To ensure low-overhead (storage) and efficient (processing) provenance collection in Century, we adopt a novel hybrid provenance model, where dependencies and lineages are asserted on individual data events, but using stream-level semantics. In particular, we have designed a *time-value centric* (TVC) provenance model, which uses the fact that the output event generated by most of the Century PEs at a particular time instant are dependent on data samples from input streams that lie within a finite, well-defined time-window (or sets of such windows). In particular, a reasonably generic, time-invariant relationship between the input and output values of a PE can be expressed as:

$$e_i(t) \Leftarrow \bigcup_{j:S_j \text{ is input to } PE_i} \bigcup_{k=1}^{L_j} \left\{ e : e \in S_j\left(t - start_{jk}, t - end_{jk}\right) \right\}$$

$e_i(t)$ is a discrete event generated by PE_i at time t—it corresponds to a stream S_i and depends on specific samples from a collection of input streams $\{S_j\}$. Also, L_j is the number of disjoint 'time intervals' for stream S_j such that $e_i(t)$ depends on data samples from S_j lying within these time intervals, with $start_{jk}$ and end_{jk} defining the boundaries of the k^{th} interval. As a practical generalization of the above time-dependent model, we allow the time-dependencies to be state (or value) dependent—in other words, a single PE may have different time-dependencies on its input streams, depending on its current 'state'.

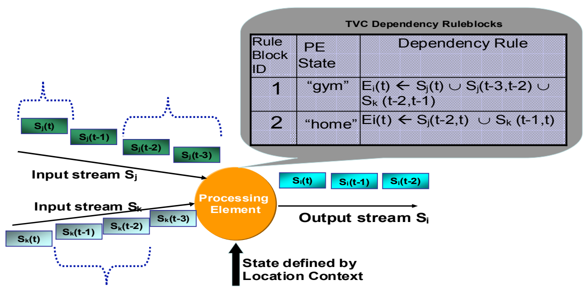

Figure 2: The PE-based model for expressing TVC provenance. The curly braces indicate the input dependencies on $S_i(t)$ for the state "loc=gym".

Figure 2 expresses the architectural model by which the Century Provenance subsystem captures the TVC-based dependencies. The Provenance subsystem associates a set of RuleBlocks with each PE instance such that each RuleBlock corresponds to a particular PE state. Whenever a new PE is instantiated, a new set of corresponding RuleBlock entries (containing the IDs of the actual streams bound to the PE) is persisted in the Provenance store. (In the figure above, the PE with id PE_i is bound to two streams with IDs S_j and S_k - the stream IDs thus provide a logical identifier for a collection of events that correspond to the same 'physical' parameter being sensed). Moreover, every data element (SDO) generated by the PE is 'annotated' with a small amount of metadata (including its creation timestamp and its stream ID); this metadata is persisted along with the actual event objects in the Event DB. It is then easy to see that the set of "upstream" data items involved in the creation of a specific data item e (possessing a timestamp, t_s, and a stream id, SID) can be resolved by first finding the corresponding RuleBlock entry in the Provenance store, and using the corresponding stream IDs to retrieve the matching data items from the Event DB. This hybrid approach is described in detail in [17] and has been shown to result in very low provenance-related storage overhead. The provenance tracking process is also computationally efficient (the Provenance store is not modified on every incoming data event) and is thus conducive to real-time stream processing.

5. Monitoring Patients with Congestive Heart Failure

We are driving the development of Century with a representative congestive heart failure (CHF) monitoring application that we are currently building. CHF is a chronic disease currently affecting a large patient population worldwide. Symptoms for it can be detected by monitoring several biometric signals. ECG monitoring can reveal several heart related problems that could result in heart failure, including heart attacks, rhythm disorders and long-standing strain on the heart from high blood pressure.

Other important event sources typically used to monitor congestive heart failure are blood pressure cuffs, detecting hypertension, and scales measuring rapid increase in body weight that could indicate abnormal body fluid retention. (Abnormal fluid retention is a strong indicator of CHF.)

Accordingly, our CHF monitoring application processes event streams collected by 3-lead ECG sensors, systolic and diastolic blood pressure cuffs and weight scales.

Figure 4 represents a PE graph consuming event streams produced by these data sources and generating CHF alerts. As sensor events enter this graph on the left side at the source PE, they are demultiplexed by type according to the Century type system described in Section 0. The top two paths subscribe to ECG readings and perform an arrhythmia analysis on these ECG readings by applying well known signal processing techniques [23]. The Frequency Analysis PE applies several band pass digital filters on ECG readings to help detect different types of arrhythmia. The QRS detector, RR detector and PT detector also apply well known digital signal processing techniques to ECG streams to extract characteristics of the different waves composing ECG signals. The Arrhythmia Alerting PE consumes these ECG waves, together with the output of the Frequency Analysis PE to generate arrhythmia alerts. These alerts are consumed by the CHF Alerting PE. This PE also consumes weight and blood pressure events to determine the CHF state of the observed patient. Note that the time scales used by the various analysis components can vary widely; as an illustrative example, arrhythmia patterns may be monitored on an hourly basis, while patterns of abnormal weight gain may be determined using a week of weight readings.

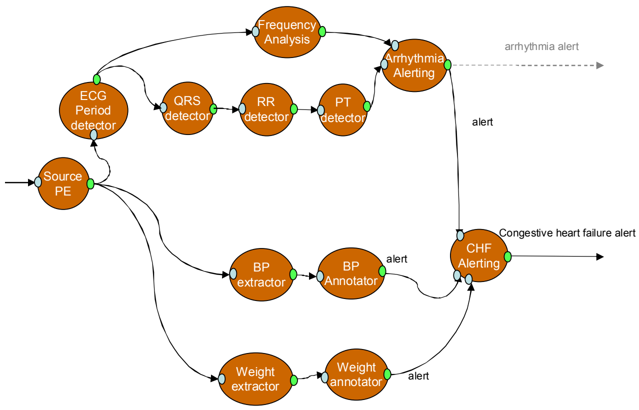

Figure 3: A Century analysis graph for the monitoring of patients with Congestive Heart Failure

6. Conclusions

In this paper we have presented an architectural overview of Century, an extensible framework for analysis of remote health monitoring data streams over a large patient population. This framework has been designed to scale to very large event rates, arising from a combination of both high-rate event streams and a large set of medical streams. In addition, it contains a provenance subsystem allowing stakeholders to inquire about the pedigree of the clinical assertions made by the system. The development of Century is currently driven by the design and implementation of a congestive heart failure application.

In the future, we plan to validate our design choices with experiments measuring the scaling capability of this infrastructure. We also plan to add additional functional requirements enabling end users to protect the privacy of the data processed in this system.

7. REFERENCES

1. World Health Organization, 2003. Global Strategy on Diet, Physical Activity and Health. http://www.who.int/hpr/NPH/docs/gs_diabetes.pdf, accessed 13 June 2006.
2. National Coalition on Health Care, 2004. Health Insurance Cost. http://www.nchc.org/facts/cost.shtml, accessed 13 June 2006.
3. Harvard School of Public Health, 2006. Diabetes. http://www.hsph.harvard.edu/nutritionsource/diabetes.html, accessed 13 June 2006.
4. Managed Healthcare Executive, 2005. "Disease Management of the Future Promises More Remote Health Monitoring." http://www.managedhealthcareexecutive.com/mhe/article/articleDetail.jsp?id=150898, accessed 13 June 2006.
5. R. Rizza, R.Vigersky, H. Rodbard, P. Ladenson, W. Young Jr, M. Surks, R. Kahn and P. Hogan. "A Model to Determine Workforce Needs for Endocrinologists in the United States Until 2020." *Endocrine Practice*, 2003, May-Jun; 9(3):210-9
6. R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," IEEE SNPA Workshop, May 2003.
7. G. Lubbe, "Wristwatch developed to combat malaria," Computer Business Review Online, January 18, 2006, http://www.cbronline.com/article_news.asp?guid=DF7D4D44-1C2F-4B61-B513-31A6DD34031A
8. L. Amini, H. Andrade, R. Bhagwan, F. Eskesen, R. King, P. Selo, Y. Park and C. Venkatramani, "SPC: A Distributed, Scalable Platform for Data Mining", Invited talk, SIGKDD 2006,Workshop on Data Mining Standards, Services, and Platforms, Aug. 20th, 2006.
9. M. Blount, V. Batra, A. Capella, M. Ebing, W. Jerome, S. Martin, M. Nidd, M. Niemi, S. Wright, "Remote health-care monitoring using Personal Care Connect", IBM Systems Journal, vol. 46, no. 1, 2007.
10. GPFS - The IBM General Parallel File System, www.ibm.com/systems/clusters/software/gpfs.html
11. M. Blount, J. Davis, A. Misra, D. Sow, M. Wang, "A Time-and-Value Centric Provenance Model and Architecture for Medical Event Streams" *HealthNet 2007.*
12. D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: A new model and architecture for data stream management" VLDB Journal, 12(2), 2003.
13. D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik, "The design of the borealis stream processing engine". In 2005 Conference on Innovative Data System, 2005.
14. A. Bar-Or, D. Goddeau, J. Healey, L. Kontothanassis, B. Logan, A. Nelson, and J. V. Thong, "Biostream: A system architecture for real-time processing of physiological signals", In 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society, 2004.
15. P. Groth, M. Luck, L. Moreau, "A protocol for recording provenance in service-oriented grids", Proc. of the 8th International Conference on Principles of Distributed Systems (OPODIS'04), December 2004.

1. K. Muniswamy-Reddy, D. Holland, U. Braun and M. Seltzer, "Provenance-Aware Storage Systems", Proc. of the USENIX Annual Technical Conference, 2006.