SENSAGE

The Event Analysis and Retention Dilemma

The critical challenge of managing and making use of ever-increasing amounts of event log data for real-world compliance audit and investigation

SENSAGE

The Event Analysis and Retention Dilemma

Introduction

Organizations, both public and private, are deploying information technology applications to yield greater efficiencies in internal operations and provide their customers and partners better service through online access. As IT systems continue to expand in size and complexity, the need to effectively monitor and manage these systems also has increased. Understanding every event taking place within these IT systems helps enhance response time, maximize availability, and lower costs—while also reducing security risks, and complying with government regulations.

What is an "event?" The atomic data unit used as a basis for system monitoring is referred to as an "event." Each component of a system generates events to signal that something of significance has happened for that component. System components include perimeter network components, internal network infrastructure components, security devices, application middleware, business applications, and databases. Streams of events are commonly referred to as logs.

It was not uncommon for enterprise event data to be selectively collected, selectively sampled, or collected but never used or maintained. Given the stated requirements for continuous process enhancement, corporate governance and compliance mandates—the management of event data can no longer be dismissed.

As such, several strategies have evolved around using event data to better manage IT systems. Initially event data was stored in log files and made available for visual inspection. Event data was analyzed by system administrators using time-consuming ad hoc methodologies such as home-grown tools and scripts. These methodologies became extremely difficult, tedious, error prone and in some cases impossible. As the volume of event data exceeded the ability to derive value from such highly manual methodologies, a variety of commercial log analysis tools were created. Initially, these log analysis tools focused on exposing web site access trends in order to improve the effectiveness of web sites for marketing and customer acquisition purposes.

As security incidents emerged as a significant IT issue, events were used to detect, analyze and prevent security breaches. The use of event data bifurcated into two methodologies. The first was to monitor the flow of events, correlate events in real time, and detect patterns indicating potential security intrusions. This was referred to as security event management or incident response. The second methodology was to store events for longer periods of times to provide historical trend analysis, investigation, compliance reporting and audit support. This comprises attributes of security analytics.

With longer-term event data storage and management, security analytics for forensic investigation and root-cause analysis becomes possible. By combining events from a broad number of system components into a central location, security staff would not need to examine multiple heterogeneous logs. This better leverages the use of a security analyst's time by filtering irrelevant information.

In addition to using event data for security purposes, event data is also being used for system management to help monitor and improve the health of a system.

The focus of this white paper frames the demands, scalability challenge and approaches concerning the management, analysis and long-term storage requirements of events for the purposes of compliance, security and system management. In this paper we discuss trends and forces that are creating requirements to manage and store increasing amounts of event data. We illustrate how and why Relational Database Management Systems (RDBMSs) were initially adopted for storing and managing event data. Then we explain the inherent limitations of RDBMSs for enabling security analysis and retention. Specifically, we will characterize aspects of event data that make it unique and distinguish it from generic business data. Finally we introduce the Sensage solution and describe the advantages of using Sensage to store, manage and analyze event data, and how it is uniquely suited to meet security, compliance and investigation requirements within gigabit-class network environments.

Why Manage More Event Data?

Companies are encountering a number of business imperatives that involve storing, managing and analyzing increasing volumes of event data. Some companies have already experienced a need to store and manage greater volumes of event data. Other companies do not have or are not yet aware of this need. For companies that currently believe they do not have a need to manage large volumes of event data, this section will either a) help confirm that conclusion, or b) help create awareness of the impending or future requirement for managing increased volumes of event data. Below we have identified and summarized the main drivers for increasing volumes of event data.

Increased Sophistication of External Security Threats

External threats are not only increasing in complexity, they are also increasing in length of duration. The time to conduct a successful attack has extended from hours, to days, to weeks, and in some cases, to months.

As companies increase their ability to prevent and detect external threats the sophistication of those posing the threats also increases. External threats are not only increasing in complexity, they are also increasing in length of duration. The time to conduct a successful attack has extended from hours, to days, to weeks and in some cases to months. The ability to detect such threats with this expanding time range is directly limited by the time range represented by the event data available for analysis. To keep pace with the ever increasing time range of attacks, security managers must have access to greater volumes of event data.

Increased Sophistication of Internal Security Threats

Internal security threats are often more serious and costly versions of external threats. The person conducting an internal threat has more information, more authorized access points, more awareness of the value of corporate assets, more knowledge of IT infrastructure and most importantly, more time. Historically, security management has been mostly focused on external threats—even though the greatest financial and legal risks come from inside. To detect internal threats, analysis of a longer time range of event data is required, and that analysis must include both authorized and unauthorized access events. As companies strengthen their capabilities to manage internal security threats, access to more event data is crucial.

Compliance with Government Regulations

The need to comply with government regulations creates a legal mandate with regard to the quantity and quality of event data that must be captured, stored and made accessible. This relationship between compliance and event data is an extensive topic, but bears some discussion here as it directly results in increased needs for event data accessibility and storage. For a more complete treatment of this topic see Sensage's white paper titled "Using Log Files in Digital Forensics and Compliance." Here, we summarize the relevant conclusions of that paper:

- There must be no time gaps in event data.
- Event data for all related assets must be available.
- All original event data must be available, which means there should be no filtering, interpretation or aggregation.
- A chain-of-custody for event data must be demonstrable

Event data can be considered possible forensic evidence or some future criminal action. Any missing or filtered event data will essentially be viewed as contaminated evidence, and therefore bring under suspicion or inadmissibility the entire set of available event data. Compliance legislation such as Sarbanes-Oxley and the Gramm-Leach-Bliley acts affect all corporations. Other compliance laws such as HIPAA and BASEL II affect specific vertical industries, while FISMA, NISPOM and DCID relate to government agencies. The scope of compliance will vary from business to business, but all companies are required to comply with some level of regulation. The legal mandate of government compliance creates a mandate for storing greater volumes of event data.

Corporate Governance

Recent high-profile court cases have shown that high level executives can be held accountable for failures in corporate governance. In many cases, pleading ignorance is no longer an acceptable defense. This trend of holding executives directly responsible for corporate wrong-doing motivates the demand for greater corporate control and visibility into corporate inner workings. Corporate complexity requires that IT be part of the corporate governance process. When event data management is limited, then only component level corporate visibility

can be achieved. With high-volume event data management centralized, cross-functional control and audit can be achieved.

Business Growth

Corporations experience business growth for a variety of reasons such as success in the market place, expansion into new markets, and mergers and acquisitions. In all cases the IT infrastructure must grow to match the expansion of business. The growth results in the addition of more system components, more types of components and ultimately, more system events. So even if a business wishes only to maintain existing system and security management capabilities, there is an inevitable increase in the volume of events it generates.

Expansion of Automation

To reap the benefits provided by IT efficiencies, enterprises must seek to automate more of their operations. Increased IT infrastructure automation also increases the volume of events produced.

Heterogeneous Growth

With acquisitions and mergers, businesses must manage the unanticipated combination and integration of disparate systems. As businesses grow, new generations of technology are integrated with legacy systems. These heterogeneous growth requirements compel the need for centralized management so a comprehensive view of these heterogeneous systems can be provided. Heterogeneous growth increases the number of actual combinations of interdependent system components geometrically. All of this results in an increased volume of events produced, and an increased need to correlate varieties of events.

Initially, the use of RDBMSs to manage event data met enterprise requirements, but as the demand to manage greater volumes of event data emerged, the limitations of RDBMSs became apparent.

Increased System Complexity

Effective system management yields optimized performance, minimum response time, maximum utilization of IT assets, rapid response to failures, and fulfillment of service level agreements. As the complexity of systems increases, these system management goals cannot be achieved by component level monitoring and restricted time span analysis alone. Additionally, event data filtering assumes that where system problems lie is known in advance, and does not allow for post-mortem analysis of unanticipated scenarios. As system complexity grows, the need to monitor the interdependency of system components and conduct analysis within longer time spans also increases. Therefore the ability to collect, store and analyze event data that spans many components for longer periods of time, increases the efficacy of system management. In regard to event data, system management and security management have redundant needs, and larger available volumes of event data have a multidimensional positive impact on corporate IT governance and effectiveness.

Increased Awareness of an Expanded Event Data Management Solution Set

As will be explained in following sections, popular solutions for managing event data have inherent limitations. Such built-in limitations have become somewhat accepted, so the expectation of what value can be leveraged from event data management has been adjusted, by some, to match the limitations of these solutions. As awareness increases of more robust solutions with quantum advances in their ability to handle volume and scale, expectations around the event data management solution set will also increase. There are many examples of this in our IT history. Until affordable PCs were available, personal productivity tools such as word processors and spreadsheets would have never been considered. Until the Internet was fully adopted, business models such as amazon.com, eBay and Google would never have been considered. So too, the availability of high-volume event data management technology will perpetuate the creation of more effective solutions that are capable of handling the growing volumes of event data produced by enterprises even more efficiently.

Example: *A Fortune 500 financial services company initially purchased a high volume event management solution to detect external security threats. As this company became aware of the reliability and scalability of their event management system, they extended their use to implement Sarbanes-Oxley compliance, and later to perform system performance analysis and tuning.*

Using Relational Database Management Systems to Retain and Analyze Event Data

To understand the mismatch between event data and RDBMS technology, it is helpful to understand unique event data characteristics.

Through more than twenty-five years of Relational Database Management Systems (RDBMS) technology development, RDBMSs have become the preferred and "safe" choice for almost all data management problems. Through this broad-based adoption of RDBMS technology by the IT industry, the data management problem-solving methodology has become primarily relational data model driven. That is to say that no matter what form the data originally appears in, the first step in creating a data management solution is to evaluate the data using the relational model. This process has proven to be successful in the vast majority of cases, so the acceptance of the relational model has become self-perpetuating.

As Security Information Management (SIM) vendors discovered the need to manage event data beyond real-time requirements, they followed the well-established practice of incorporating event data into the relational model and using RDBMS technology to store and analyze it. This allowed SIM vendors to focus on other areas of concern such as real-time event correlation, mitigation and user friendly presentation.

Initially, the use of RDBMSs to manage event data met enterprise requirements, but as the demand to manage greater volumes of event data emerged, the limitations of RDBMSs became apparent. To understand why RDBMS technology now presents barriers to meeting event data management requirements, it is helpful to understand the foundational requirements that drive RDBMS technology. These requirements are listed and summarized below.

Transactional Data

RDBMSs are designed to support the commit/rollback protocol which follows the standard that only complete transactions (data changes) will be permanently stored and visible. Well-designed applications have very small transactions that take microseconds to complete. Any data stored in an RDBMS database can be changed within a transaction. To support this, RDBMSs have elaborate logging sub-systems that log every change in order to be prepared in the event a transaction rollback occurs.

Isolated Concurrent Access

RDBMSs present a virtual view of the data, so a given user only sees committed data, or data that the user has changed. Although other isolation levels are supported, the concurrent isolation paradigm requires synchronicity and locking sub-systems at the row level.

Infrequent Schema Changes

Before data can be loaded into an RDBMS, a schema which defines the semantics and existing data relationships must be in place. This creates a requirement that a data model must be complete before an application can be created. Relational schemas are generally static, or evolve slowly. Infrequent changes in application requirements drive schema changes.

Precision Queries

RDBMS are designed to optimize precision queries on structured data. This means that precise information is known about the data before a query is formulated and the data itself has been structured to fit into a predetermined model or schema. Often queries are statically stored and optimized, with the query data being variable. RDBMSs are optimized best for unique key queries such as customer number or invoice number. RDBMSs are sub-optimized for range or pattern matching style queries. Examples of this would be to "list all invoices over $100,000" or "list all companies with '.com' in their names." These types of queries usually involve a complete table scan. RDBMS databases must be tuned to support a specific set of applications. The tuning is accomplished through indirect references to data such as indices, or by data organization such as data clustering. Therefore once a database is tuned for a specific set of applications, it can easily become suboptimized for other applications.

Unique Characteristics of Event Data

With a clear understanding from the previous section of the dominant design goals of RDBMS technology, it is important to understand the nature of event data next. Through understanding the objectives of RDBMS technology and the characteristics of event data, an analysis can be performed showing how well event data maps onto RDBMS technology. These characteristics of event data are listed and summarized as follows.

Non-transactional

The requirements for event data are that it be stored, searched, removed and archived. Once event data is stored, it will never be updated. In fact, for compliance purposes, altering and deleting event data should be strictly prohibited. Although there are some transactional semantics for event data, they are not the same nor as stringent as those for relational data.

Time-based

Event data is a collection of data about a particular event, at a specific point in time. This means that every event will have a time stamp associated with it. Any search on event data is likely to have some time boundary.

Field Repetitiveness

Event data is generally highly repetitive. For example, a company will have a relatively small set of authorized users. Event data for successful connection events will repeat this small set of authorized users over and over again. Other examples of highly repetitive data are URLs, IP addresses and IDS signatures. Read-only databases disable transactions using traditional RDBMS.

Time-based Archival or Removal

All event data is eventually removed or archived based on aging, as determined by system or security management requirements.

Variable Search Requirements

Searches on event data can be precise or pattern oriented. For example, one search may require a precise matching of a user name while another search may be based on patterns found in a URL such as “hotmail.com.” Although both kinds of searches must be supported, most log data is unstructured and must be searched using some form of pattern matching. Therefore, event data storage must not be organized to optimize precision searches at the expense of sub-optimizing pattern-based searches. The unstructured nature of event data is resistant to query optimization via the creation of indices.

Evolving Search Requirements

The event data search requirements evolve with the security and systems management landscape. New searches must be created to detect newly discovered security threats or to monitor new system components. This can happen on a daily or weekly basis. During a forensic process, the results from one search will determine the nature of the next search. Because of the unpredictable nature of rapidly evolving search requirements, it is insufficient to have an event storage system optimized for current requirements at the cost of sub-optimization of future search requirements. This is contrasted with RDBMS databases which have query requirements which are static or evolve slowly with the gradual introduction of new application requirements.

Near Real-Time

Event data is created in real time and must be loaded at least as fast as it is created. Although all event data does not need to be available in real-time, load rate must keep pace with creation rate in the long run. Unlike relational data, the load rate cannot be slowed down by reducing user response time. System components create event data at a rate based on their usage, independent of the event data load rate.

Data Protection

Event data must be protected from being changed or destroyed from any source, including applications, users, and component failures. Destruction or modification of event data can result in non-compliance with government regulations, increased security risk, or failures in system management. Although relational data has similar data protection requirements, in many cases these requirements are not as absolute, and protection failures don’t have the same pervasive impact.

System Wide High Availability

Security and system management designed with the use of event data become reliant on the event data. Should the event data become unavailable, operations may have to be shut down to prevent security breaches or noncompliance. In contrast, loss of availability of relational data may stop the operation of a group of applications or segment of the system, yet allow other independent operations on the same system to continue.

Contrasting Relational Data with Event Data

In the previous sections we exposed the nature of event data and described the dominant design objectives of RDBMS technology. Through this you can begin to see that event data and RDBMS technology are at best, poorly matched. In this section we will show a side-by-side comparison of standard relational data and event data. This comparison illustrates the stark contrast between these two kinds of data:

Area of Concern	Event Data	Relational Data
Transactions	Limited transactional requirements	Fundamental requirement
Isolation Concurrency	Stored data is never updated so all stored data is available to all users (subject to authorization filtering)	Must present an isolated virtual database view to prevent visibility of non-committed data
Schema	Must be general and flexible to accommodate future event types. Semantics of data are often determined at search time	Generally static and must be determined before data can be stored and accessible by applications
Search	Search criteria can be precise or pattern based and search requirements evolve rapidly	Search criteria are precise and databases are optimized to support a known set of queries Search requirements are static or evolve slowly
Time Attribute	Time attribute is part of every event and is usually a key criterion for search	Time is one of many possible attributes and may not be present
Life Cycle	All event data is eventually removed or archived based on aging and retention rules	Has no consistent life cycle requirements Most data has an indefinite life span
Data Protection	Data protection to prevent destruction and modification is absolute and must be supported regardless of user access or component failure Data protection failures may cause compliance failures	Data protection is customized per application and is based on database security authorizations and application business logic
High Availability	Lack of availability is likely to impact the entire system and may have legal ramifications	Lack of availability may only impact a segment of the system
Load Rate	Load rate must keep pace with event data creation rate System components generating event data do not wait for event data loading	Load rate is based on user response time Increased response time reduces the load rate requirements

Storage Barriers of RDBMS Managed Event Data

Previous sections show a fundamental mismatch between event data management requirements and RDBMS core technology strengths. The mismatches can be roughly categorized as either a) RDBMS over-head not required to manage event data or b) lack of technology to support the unique requirements of event data management.

Industry experience and empirical evidence show that these are not theoretical mismatches, but are manifested in real failures to meet compliance, security and system management requirements. These failures can result in failing to meet audits, increased security risks, and increased IT infrastructure costs.

Industry experience and empirical evidence show that these are not theoretical mismatches, but are manifested in real failures to meet compliance, security and system management requirements.

Problems with using a traditional RDBMS to manage event data are multi-faceted, and sites using RDBMS technology in this manner are likely to experience most, if not all of these problems. Below we have identified each of the “barriers” created by using an RDBMS for event data.

Increased Storage Requirements

To support the commit/rollback protocol, RDBMSs maintain a transaction log that will allow for the potential rollback of non-committed updates. To optimize precision searches for specific applications, RDBMSs support the creation of special access data structures such as b-tree indices and hash tables. Some RDBMSs support isolation concurrency through the storage of lock information along with the data. To support data versioning, multiple versions of the data are stored.

This is just a partial list of RDBMS features the result in upward pressure on storage requirements. The net result of this impact is that the volume of RDBMS storage overhead may be up to three times larger than that of the original data, resulting in a 4 to 1 expansion of storage requirements.

Increased CPU Requirements

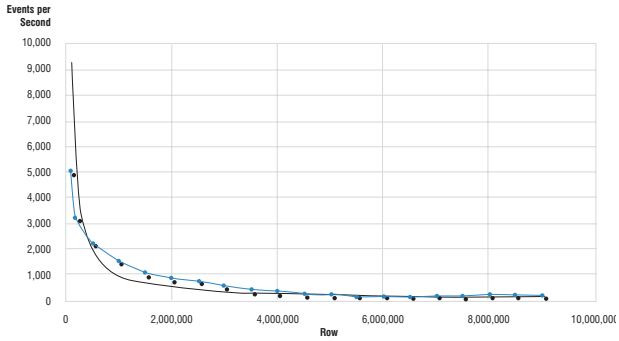
Row-level concurrency control requires that an RDBMS check a lock/transaction table for every row access. Most RDBMSs use a paging system to allow multiple transactions to share data and reduce I/O requirements. These and other internal structures use additional CPU cycles.

Increased I/O Requirements

The increased storage requirements result in increased I/O requirements. Indices must be maintained in tandem with every row update.

Geometrically Decreasing Load Performance

As the volume of data increases, the cost and time to load the next row of data increases geometrically. This is due to the maintenance of indices. RDBMS are optimized for the precision searching of data which is loaded one transaction at a time, and not for bulk data loading. The graph below shows the dramatic decrease in load rate as the volume of events increases.



Geometrically Decreasing Load Performance

Load rate decreases to fewer than 300 events per second when the event volume reaches 2 million. Due to the nature of RDBMS indices, this degradation in event loading rate is permanent. Although the absolute numbers may vary based on hardware configuration, the shape of the curve will be the same.

Search Specific Optimization

Search is optimized in RDBMSs by creating indices which anticipate a specific set of search criteria. For example, if it is anticipated that many searches will involve a user name, then an index will be created on the user name column. Searches which fall within the bounds of the anticipated search criteria will execute quickly. Searches that do not fall within the bounds of the anticipated criteria will cause a complete table scan.

Some common event data searches cannot be optimized by an RDBMS at all. These include substring and pattern matching searches. These also result in a complete table scan. For large databases of billions of rows, these searches can become multi-day queries rendering a result of no practical value.

Example: *A customer attempted to perform a search against 1 billion web proxy events that would return timestamp, IP address, User ID and URL for all records where the URL contained a specific string. Because this is a pattern search, the RDBMS was required to conduct a full table scan and could not take advantage of any indices. This resulted in a search that took days to complete using high-cost, massive multi-processor, high-performance servers.*

Search Optimization and Load Rate Trade-off

To increase the number of search scenarios that are optimized, more indices must be created. The more indices that are created, the slower the event data load rate becomes. This puts the DBA in the position of constantly maintaining a balance between search performance and load performance. The process of finding the load/search balance requires indices to be created and dropped. Search scenarios must be anticipated and prioritized. If the event data load rate becomes unacceptable, then indices must be dropped. Dropping an index can reduce search performance in ways that are hard to predict. For large volumes of event data creating and dropping indices are major operations that can temporarily shut down database operations and take hours or days to complete.

Faced with the limitations imposed by RDBMS solutions for event data management, SIM companies and their customers have adopted a number of strategies to mitigate RDBMS shortcoming.

Mitigating RDBMS Event Data Management Barriers

Faced with the limitations imposed by the RDBMS solutions for event data management, SIM companies and their customers have adopted a number of strategies to mitigate RDBMS shortcomings. These strategies represent a valiant and costly effort to deal with the inherent failings of RDBMS technology in managing event data. If the sum of all of these strategies were successful, then this paper could end in this section. However, each strategy used is either insufficient, risk-producing or both. The following describes popular strategies that have been used, and identifies the failings of the strategy including the potential of increased risk.

Data Filtering

To reduce the amount of event data that needs to be stored, filters are created to reduce both the number of events and the amount of data stored for each event. While this strategy allows event data storage to span longer ranges of time, it creates harmful side effects. Filtering pre-supposes that the nature of searches needed in the future is known in advance. Unanticipated security or system management scenarios may require data not stored, rendering limited value from any event data that is stored.

Example: *Denied access is monitored but successful connections are not. If a buffer overflow attack is being accomplished by a server making excessive outbound FTP connections, examining the event data collected will fail to identify the attacking server.*

Additionally, government compliance requires that all original event data be available to establish full context and to ensure that there was no data tampering. In general, data filtering produces an incomplete record resulting in limited value for the event data collected.

Limited Time Range Searches

Storage capacity directly impacts the searchable time range of events. Searches are artificially limited in time scope based on storage capacity. This approach ignores business imperatives that require longer-term search capability.

Example: *For a particular company, only one week of event data is kept. A sophisticated attacker spreads pre-attack reconnaissance over a few weeks. Use of the available event data is unable to detect this low-and-slow attack.*

Limited Component Monitoring

The requirements for event data storage can be decreased by reducing the number of system components that are monitored. This requires an analysis to determine which components do not need to be monitored and, as in data filtering, pre-supposes the nature of future event analysis. Discovery of new security and system management

scenarios may expose the need to have captured and stored event data from non-monitored components. This would create a "missing link" that would impede a forensic investigation from uncovering the root cause of a security breach. Compliance mandates typically require comprehensive monitoring of specified components within IT infrastructures.

Example: *A hospital uncovered a risk scenario related to leakage of its VIP patient information via web surfing from shared workstations which are also used for patient data access. To determine the root cause of the leakage, daily event data from windows logins, web proxies, DHCP and a patient management application needed to be correlated for a trailing week. However, collection of web proxy monitoring event data was previously eliminated to reduce RDBMS storage requirements, making discovery of the source of this critical scenario impossible.*

Event Storage Limited by Capacity

The amount of event data that is stored can be limited by the available RDBMS capacity in terms of storage space and load rates. When event storage reaches a pre-defined threshold, the oldest events are purged until event storage is below the threshold. This strategy guarantees control over the amount of data stored, and load rate achieved, at the sacrifice of a predictable time range for available event data. A spike in activity would effectively

Example: *To catch low-and-slow attacks, the event storage policy is changed from a one-week retention period to a three-month retention period. The RDBMS capacity is increased by 1,200% by purchasing additional disk capacity. But the event data load rate declines to the point where load rate cannot keep pace with event data creation rate. Moreover, the increased capacity does not hold three months of event data because of the unanticipated non-linear increase in space required for indices. Net result, the expected time span of collected data is not achieved, and the low-and-slow attacks still remain undetected.*

Two-tier Storage Architecture

To alleviate the high cost of RDBMS storage, aged events can be removed from the database and archived into lower cost compressed storage. Should events from the archive be needed, they must be uncompressed and restored to the database. Removal and restoration of event data from an RDBMS database are time consuming, creates resource contention with other operational data loading, and are potentially manual operations requiring DBA and system administration resources. Also, compression algorithms are not sensitive to the repetitive nature of event field data and therefore only achieve standard compression ratios. While a two-tier strategy is a good approach for Information Life-cycle Management (ILM) it is no substitute for adequate on-line event data storage.

Time-based Database Segregation

To mitigate geometric event data loading performance degradation, event data can be segregated into separate databases based on time ranges. This effectively creates a meta-index based on event time that is maintained by the user. This has the advantage of creating a sustainable minimum event data loading rate. However, with this strategy, part of the search optimization burden now shifts to the user. The formulation of searches becomes more complex and must consider which databases should be searched. What used to be a single search must be manually broken up into multiple searches and the results must be manually aggregated.

Evolution of Storage

In previous sections of this paper, we document how RDBMS technology was developed primarily to address business data which is transaction and record oriented. We identified unique characteristics of event data, and described the differences between relational data and event data. We then demonstrated how RDBMS technology is unable to adequately address the unique characteristics of event data, and how attempts to force an RDBMS solution for event data management often create negative ramifications, such as significantly diminished performance and increased storage requirements. We explained some of the strategies that have been used to mitigate RDBMS shortfalls, and how these strategies serve to produce even more issues.

We conclude that there is a clear need for an event storage technology that overcomes RDBMS shortfalls and leverages the unique characteristics of event data. This technology would not be competitive with RDBMS technology, but would instead follow recent trends of purpose-built storage solutions. There are three distinct generations that comprise the evolution of storage technology:

- **First Generation: Direct Attached Storage** First generation architectures consisted of a hard disk attached directly to a computer, with storage controlled by the computer's CPU. Today, greater than 95% of all computer storage devices (disk drives, disk arrays, and RAID systems) are directly attached to a computer through various adapters—via standardized software protocols such as SCSI, Fibre Channel and others. This type of storage is alternatively called captive storage or server attached storage.
- **Second Generation: *Storage Architectures*** As architectures evolved, they began utilizing a combination of computers, network, and DAS that essentially provide virtual DAS for client computers. Specifically this refers to Network Attached Storage (NAS) and Storage Area Networks (SAN.)
- **Third Generation: *Data Type Specific Storage Architectures*** Third generation storage architectures are similar to generic data storage architectures, but are specifically designed to manage the unique characteristics of the type of data stored.

We conclude that there is a clear need for an event storage technology that overcomes RDBMS shortfalls and leverages the unique characteristics of event data.

The Sensage Solution

Sensage delivers a high performance, scalable means for organizations to centrally aggregate, cost-effectively store, dynamically monitor and efficiently analyze massive volumes of event log data over long periods of time while retaining the original source data.

Sensage eliminates the standard RDBMS overhead not required to manage event data, and materially increases the performance and capacity to manage massively large volumes of event data.

Our solution provides significant benefits to customers needing advanced event data management.

High Performance Search

Execute searches in minutes or hours, where RDBMS searches often take hours or days.

Event Data				
First Generation	Standard Spindle Storage	DAS	Direct Attached Storage	IBM, Seagate
Second Generation	Storage architectures	NAS	Network attached storage	Network Appli-ance, IBM, HP, Second Maxtor, Quantum, EMC…
		SAN	Storage area network	HP, IBM, Hitachi, EMC…
Third Generation	Data type specific storage architectures		Content addressable storage	EMC…
			Email archive storage	EMC, Veritas/ KVS…
			Security event analytics	Sensage

High Volume Loading

Data loading keeps pace with enterprise-wide event collection for gigabit class networks, with no degradation based on the volume of data stored.

High Volume and Low Cost Storage

Use low cost Linux servers to store highly compressed data. No expensive RDBMS licenses are required. Servers are more efficiently utilized due to the elimination of RDBMS overhead.

Low Cost of Ownership

This solution requires no DBA. Data organization is simple and self-tuning.

Incremental Scalability

Adding servers enables capacity and throughput to be scaled to match business growth. Scales proportionately to the number of servers added.

High Availability

Built in redundancy allows continued operation even with a server failure.

Data Protection

Event data is protected against modification by all outside sources. Data redundancy protects against loss of data in the event of component failure.

Sensage Architecture

In this section we provide a high level discussion of the architectural elements which enable Sensage to deliver scalable high capacity event storage.

Our core technology is a combination of:

- Server clustering (MPP architecture)
- Data compression
- A non-transactional model, and
- Seamless access to online and archived data in a single query

Clustered Parallel Distribution

In this section we describe the ways that Sensage leverages clustered server architecture to distribute workload and achieve parallel computing on a massive scale.

Near Real-time Loading

Event data is created in real time and must be loaded as fast as it is created. To address this requirement, Sensage has a “trickle-feed load” feature, loading and making data available for querying near real-time. This is done through special data structures that capture the near real-time data and make it available for querying before it is merged into the actual columnar data store.

Distributed Loading

As data is loaded into the Sensage solution, it is evenly and concurrently distributed across all the servers in the cluster. This maximizes load efficiency.

Distributed Search

Search requests are also evenly distributed across the Sensage servers. Each server conducts its portion of a search in parallel with the other servers. The final results from each server are aggregated and returned to the user.

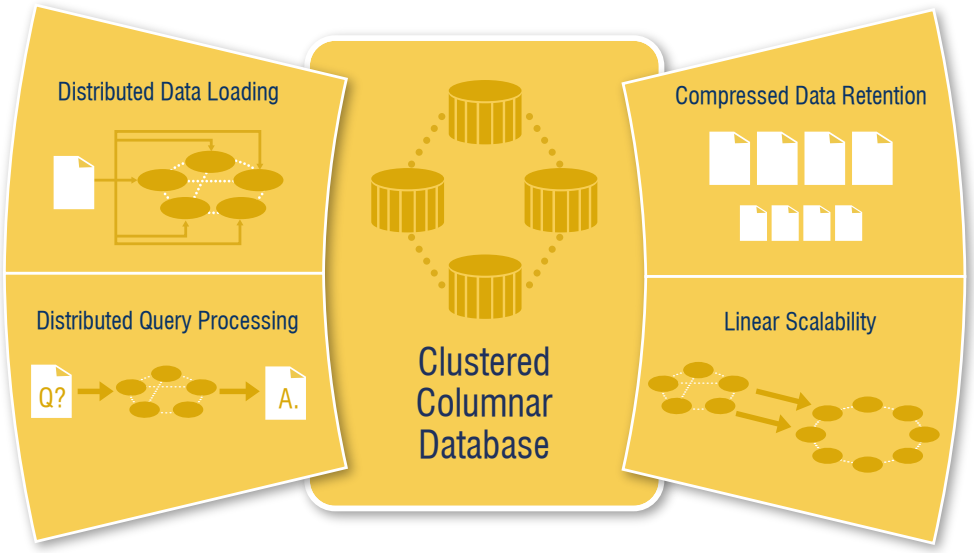
Distributed Aggregation

Aggregation searches that use 'GROUP BY' are distributed across the Sensage servers for complete parallel processing of the aggregation operation. The results of each server are then aggregated into a final result.

Data Redundancy

Every event is recorded twice in the Sensage server cluster. Each copy is stored on a separate server. Should a server fail, the server that holds the copy of the failed server's event data automatically takes over all search operations for the failed server.

Searches and loading continue with throughput degraded by a factor proportional to the number of servers in the cluster. For example, in a five server cluster throughput is reduced by 20% should a single server fail. This design enables the Sensage architecture to provide high availability with marginal loss of performance



Key Sensage architectural elements

Event Data Specific Storage Organization

Sensage's storage organization is specific to the unique nature of event data, and produces significant advantages when managing that data.

Time-based organization

The data on each server in a Sensage cluster is partitioned in time ranges. This creates advantages at load time and search time. For load time, since event data generally has increasing time stamps, the likelihood of combining new load data with data already loaded is small. This dramatically reduces data reorganization needs. For search with time constraints, our search engine quickly eliminates the need for scanning data that does not meet the time constraints.

Column-based Compression

Within our time-based organization, data is placed into columnar storage and then compressed when written to disk. High compression ratios are achieved because of the repetitive nature of event data within a column. Compared to the volume of data stored in an RDBMS database, SES achieves up to a 40:1 compression ratio. (For a further detailed comparison see Appendix A.) Only the columns referenced in a search are decompressed and searched. As with time-based organization, this eliminates the need to decompress and scan large volumes of unnecessary data. The native storage format of our Sensage solution is compressed, with decompression only required after the event data has been selected based on time or column references. Because the basic unit of storage is a flat file, data removal and archival operations are simplified and extremely fast.

No Indices

Because the unstructured nature of event data indices renders little value, Sensage delivers dramatic search response time improvement through distributed parallel searching, and event specific data organization. And, unlike an RDBMS, Sensage architecture requires no indices therefore there is no need for a DBA resource to create and drop indices to balance between search and load performance. There is also no overhead of index maintenance during loading for Sensage. This means that the event data load rate will remain constant, no matter how much data has already been loaded. Additionally, because no indices are needed, there is also no need for storage of index information. This substantially reduces storage requirements as compared to those by RDBMS-based SIM products.

Non-transactional Model

Sensage delivers unparalleled performance versus RDBMS-based SIM products, largely because of its nontransactional model. This is accomplished by minimizing overhead and optimizing use of computing resources.

No Concurrency and Locking Overhead

Because event data is never updated, our solution has no RDBMS overhead of row and table locking. Searches never need to wait for updates.

No Transaction Log

Since the commit/rollback model is not meaningful for event data, the Sensage solution avoids CPU, I/O and storage capacity overhead required to maintain a transaction log.

Access to data via 3rd party Business Intelligence Tools

Sensage provides the first and only SIEM solution that supports an open access interface to event data using database connectivity (ODBC/JDBC) APIs. These APIs enable any third party Business Intelligence tools to easily integrate with the Sensage SIEM and log management solution. Sensage has created a purpose-built architecture supporting security event analytics which first, eliminates standard RDBMS overhead not required to manage event data, and second, implements technology specifically designed to provide scalable event storage for massively large volumes of event data. The Sensage solution provides significant benefits to customers requiring advanced event data management. Opening Sensage's security data warehouse to established BI tools enables faster, better, deeper analysis, enabling Sensage customers to extend the investment and knowledge they have in their BI tools to gain additional insight and knowledge about their security environment and broader IT infrastructure.

Sensage open access interface gives enterprises the flexibility to use business intelligence tools they already know and work with everyday. The end result is complete freedom to analyze event data, ranging from statistical trend analysis on high impact metrics to executive dashboards that summarize operations effectiveness to cost/benefit analyses on new investment decisions. This is an unprecedented capability for experienced security professionals.

Unlimited Scalability

Many Massively Parallel Processing databases can only scale to a single cluster and to date, the largest MPP cluster size in the world is 96 nodes. Sensage has solved this issue by distributing data evenly across multiple clusters and returning results from single query that spans multiple clusters. Sensage users can deploy federated deployments (multiple clusters) as needed and access the data across multiple clusters without compromising on load speed and query speed.

Conclusion

In this paper we have shown how business drivers for security risk management, system management, and government compliance have all created a need to store and manage system event data—and that the volumes of that event data is steadily increasing. Initially SIM vendors adopted RDBMS technology as a preferred event management solution. As the volumes of event data grew, RDBMS technology could not meet event data management requirements.

Sensage recognized the need to manage the tremendous volume, on-going mass and unpredictable use of event data in future analytics. From the start, Sensage designed an event-centric, high-performance architecture that is able to collect an enormous amount of complete log data, at high velocity. This robust data management solution correlates log sources virtually at query time, and provides flexibility to support a broad number of sources. It enables unparalleled precision and long-term search and trending, while significantly saving on storage capacity. Furthermore, clustering technologies provide our customers incremental scalability on load and query throughput, as well as data redundancy and capacity.

Sensage delivers a high-performance, scalable solution for organizations to centrally aggregate, cost-effectively store, dynamically monitor and efficiently analyze massive volumes of events over long periods of time, while retaining the complete original source data. This empowers organizations to respond to business threats, conduct thorough investigations, and fortify broad audit compliance processes.

About Sensage

Sensage®, Inc. helps organizations collect, store, analyze and interpret complex information to identify new threats, improve cyber-security defenses, and achieve industry and regulatory compliance.

Sensage serves our customers' most advanced Security Information and Event Management (SIEM), log management, Call Detail Record (CDR) retention and retrieval and Continuous Controls Monitoring (CCM) use cases. Hundreds of customers worldwide leverage patented Security Intelligence solutions from Sensage to effectively identify, understand and counteract insider threats, advanced persistent threats, cyber threats, fraud and compliance violations.

Combining powerful data warehousing with scalable, clustered multiprocessing and robust analytics, Sensage solutions handle all event data types, scale to petabytes, minimize storage costs and perform sophisticated data analysis. Sensage has achieved Federal Common Criteria and is the process for FIPS 140-2 Certification. Sensage partners include Cerner, Cisco, EMC, McAfee and SAP. For more information, visit www.Sensage.com, follow us on Twitter: @Sensage, and watch for us on www.youtube.com/Sensagetv.

Appendix A
Storage Requirements Comparison

In this appendix, we compare the storage requirements of Sensage SEA and an RDBMS database. Comparisons are based on the following assumptions:

Average Original Event Data Size	150 bytes
Working Days per Year	260 days
SEA Compression Ratio	10:1
RDBMS Expansion Ratio	4:1

The expansion ratios listed above for RDBMS storage are conservative when compared to those where a greater number of indices are involved. The following table compares storage requirements in gigabytes based on these assumptions for three different sizes of companies:

Company Size	Millions of Events Per Day	RDBMS Daily Storage Requirements	SEA Daily Storage Requirements	RDBMS Annual Storage Requirements	SEA Annual Storage Requirements
Medium	15	9.0	0.2	2,340.0	58.5
Large	150	90.0	2.3	23,400.0	585.0
Global 500	1,000	600.0	15.0	156,000.0	3,900.0

This table illustrates the dramatic difference in storage requirements for Sensage. For a company that generates a billion events per day, RDBMS technologies would require over 150 terabytes of storage. This is well beyond the capability of current technology, and would be prohibitively expensive. On the other hand it would take about 4 terabytes of Sensage storage to store a year's worth of events for this Global 500 company, which is well within the technological and cost parameters of such a company.

SENSAGE

Sensage, Inc.
1400 Bridge Pkwy.
Suite 202
Redwood City
CA 94065
www.sensage.com