Glassbeam White Paper

Build versus Buy — Big Data Analytics on Machine Data

glassbeam™

Contents

Build vs. Buy: Big-Data Analytics on Machine Data

Many analytics projects involving log files focus on operational event data. IT use cases around such projects typically focus on locating errors, warnings, and critical event information within mountains of data.

However, software applications and technology devices produce much more machine data than just log files. They also generate product usage capacity and license information, configurations and settings, and other business data. Like the data in log files, this data is text-based and has structure. However, it does not have consistent formatting over time.

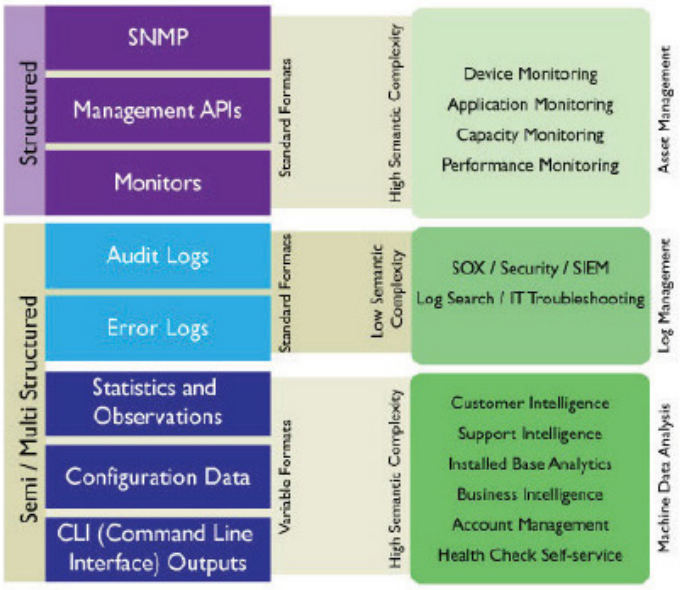
While structured event log data is generally used for data center management and IT, semi-structured or unstructured data is intended for use by support and product/engineering groups. This data therefore represents a valuable source of critical business intelligence that is too often overlooked.

Log analysis tools in the market today have evolved to process event data row by row. But to process more complex logs, companies typically choose to build their own solutions, because simple event log processing does not work for complex product logs. But the analytic-tools marketplace is changing. This paper outlines the value of using a commercial tool rather than attempting to solve the problem in house.

The challenges of in-house machine data analytics

Organizations that want to search and mine the information contained in machine data have two choices: obtaining a purpose-built analytics system, or building one in house. While the latter option may sound appealing at first, here are some of the

challenges facing organizations that want to build their own machine data analytics solutions.



Log types and corresponding market applications

Different users have different requirements

A machine data analytics solution must satisfy the requirements of a wide range of internal consumers. A support engineer working a case needs to be able to see patterns of events and statistics over time grouped by specific system components. An account manager, professional services engineer, or sales rep responsible for an account has very different data analysis needs that involve being able to quickly spot reliability, performance, and other issues involving an account. A marketing or program manager has yet different needs for spotting feature adoption and trends in product demand.

Every product has a complex and unique representation for machine data

For a set of data to be useful underneath machine data analytics, it needs to communicate detailed information about the configuration, events, and statistics of components of each product's unique architecture. Applications, appliances, software, and other components always log events, counters, and information about

internal component states, down to the level of vendor-specific abstractions. Much of this data will be vendor- specific and will not conform to a common information model for interoperability or end use. It will be largely undocumented and without formal grammar.

A useful data archive will be “Big Data”

To be of maximal benefit to all consumers, a good set of machine data needs to contain “everything.” Trending and field analysis require detailed parsing for all systems continuously, not just when problems are detected. A successful product may have thousands, hundreds of thousands, or even millions of devices or systems reporting data back regularly. The volume of data received and retained in such a case is likely to be in the range of hundreds of terabytes or petabytes over the course of a year.

Data formatting and semantics will change quickly and without notice

For machine data analytics, requirements focus on quick adaptation, edge-case coverage, and continuous business leverage. A machine data analytics solution cannot expect schematized or specially formatted data. And it needs to adapt quickly to changes in format or information from machine logs while maintaining semantic continuity with existing tools.

In-house solution: The bottom line

An in-house machine data analytics solution is a complex, high-performance big data project with associated BI tools that requires a variety of committed resources for an extended period of time. It is inherently time-consuming and risky if not planned properly with appropriate resources needed to not just design and implement, but also to maintain and manage its life cycle on a continuous basis. With a best-in-class solution like Glassbeam that is specialized for machine data analytics, ROI is far higher when one factors all the costs associated with building an in-house solution.

Glassbeam and machine data analytics

An alternative to building a data analytics system in house is Glassbeam, a cloud-based analytics solution that provides business insight for sales, support, and engineering organizations by mining machine log data.

Glassbeam is a revolutionary innovation in the large-scale analytics of semi-structured data, making it an excellent fit for a machine data analytics platform.

The core technology of Glassbeam consists of the breakthrough Semiotic Parsing Language (SPL, an intuitive declarative language along the lines of a DSL (domain-specific language). Using SPL, an analyst can describe the structure and semantics of a class of documents without needing to be concerned about specifying a program, database, or UI. For semi-structured data, this means taking advantage of the structure inherent in the data with-out any pre-defined grammars, and reflecting this structure in a high-performance SQL/NoSQL data store.

Case Study: Build vs. Buy

To compare the results of an in-house effort with Glassbeam's capabilities, let's start by looking at an example of a customer where Glassbeam was chosen as the preferred solution over internal development efforts.

At this customer site, the in-house team had been working to deliver machine data analytics value to the business for more than a year before they purchased the Glassbeam product. During this time, the project had been staffed with 2-4 people at any given time, with multiple dedicated computer and storage servers for their use. They had clearly been paying significantly more for their in-house effort than they are currently paying for Glassbeam.

From initial time when Glassbeam team was engaged to do a POC, there was little in the way of a comprehensive machine data analytics roadmap in-place. The team was pushed and pulled according to which individuals were experiencing hot issues, whether from the infrastructure design, or log data types, or queries being asked by internal business users (like support, product management, sales etc). In contrast, Glassbeam came in with a comprehensive solution from day one.

To compare the results of the in-house effort with Glassbeam's capabilities, let's start by getting a glimpse at the sort of machine data we were given to operate on. In this case, the data arrived at the product company's data center daily from each installed product in the field. It amounted to about 100 GB per month. Here are just a couple of sanitized snippets:

```
X-bundle-location: unknown X-bundle-time: 1184839212 X-autosup-
port-type: autosupport
Message-ID: <SJCEXFE039n1xxWn9R600000df9@mail.xxxxxx.com>
X-    OriginalArrivalTime: 19 Jul 2007 10:00:12.0954 (UTC) FILE-
TIME=[9948EFA0:01C7C9EB]
========== GENERAL INFO ==========
GENERATED_ON=Thu Jul 19 03:00:03 PDT 2007 VERSION=xxxxxxx
999.5.0.0-47161 SYSTEM_ID=5FP4144010 MODEL_NO=xxxxx
HOSTNAME=rig22.xxxxxxxx.com LOCATION= ADMIN_EMAIL= UPTIME=
03:00:03 up 20 days, 13:54, 2 users,
load average: 0.14, 0.06, 0.01

========== SERVER USAGE
Resource Size GiB
------------------ -------- -------- --------- ----
/backup: pre-comp - 10.0
==========
Used GiB Avail GiB Use%
- -/backup: post-comp 677.3 0.8 676.5 0% .........
-
Current Alerts
--------------
Alert Time Description
---------------- ------------------------------------------------
-----------
Thu Jun 28 13:09 Encl 1 (5FP4144010) Disk 9 is absent and should
be replaced
Thu Jun 28 13:09 Encl 1 (5FP4144010) Disk 10 is absent and should
be replaced ..........

========== MESSAGES ==========
Jul 14 18:00:01 rig22 xxsh: NOTICE (tty=<>, session=15341,
host=<>) root: command “filesys show space 2”
Jul 14 18:00:01 rig22 xxsh: NOTICE (tty=<>, session=15347,
host=<>) root: command “filesys show space 1” Jul 14 18:00:01 rig22
xxsh: NOTICE (tty=<>, session=15348, host=<>) root: command “sys-
tem show hourly-status”
Jul 14 18:00:01 rig22 logger: at 18:00:01 up 16 days, 4:54,
4395830 NFS ops, 0.1 GB data col. (0%)
Jul 14 19:00:02 rig22 xxsh: NOTICE (tty=<>, session=17849,
host=<>) root: command “filesys show space 2” Jul 14 19:00:02
rig22 xxsh: NOTICE (tty=<>, session=17855, host=<>) root: command
“filesys show space 1”
Jul 14 19:00:02 rig22 xxsh: NOTICE (tty=<>, session=17856,
host=<>) root: command “system show hourly- status”
Jul 14 19:00:02 rig22 logger: at 19:00:02 up 16 days, 5:54,
```

```
4395848 NFS ops, 0.1 GB data col. (0%)
Jul 14 20:00:02 rig22 xxsh: NOTICE (tty=<>, session=20355,
host=<>) root: command “filesys show space 2” Jul 14 20:00:02
rig22 xxsh: NOTICE (tty=<>, session=20361, host=<>) root: command
“filesys show space 1”
```

As you can imagine, the data was largely semi-structured. Not only that, there were 80+ such differently formatted sections, with formats for each differing over 100+ releases in the field. Source content included configuration files, event logs, and statistical dumps for all objects and relationships deemed important for support debugging purposes.

Glassbeam solved the above problem by doing the following key things:

- a unified source data model on the above disparate sectional data
- a unified object data model by parsing and associating things like properties of disks, adapters, shelves, etc.
- a unified relational data model for faster query performance and semantic representation

Despite variations in content and semantics over the lifetime of the product, we were able to generate a single data model, which naturally captured the attributes, component relationships, and semantics for all of them. In contrast, the in-house project needed if-then-else criteria pre-designed for each downstream query.

With Glassbeam, now this customer could ask any question in real time, such as “did this system, or any of its components, undergo a configuration change at any point in time?”. This was possible only because Glassbeam solution modeled the relationships among all components as either temporary or permanent, tracking them, allowing for different attribute classes (configuration vs. status, for example), and storing them differently for optimization according to their cardinality. Every event or statistic was associated with the object(s) with which it was related in order to answer questions like “show me the media error rate by drive model over the whole install base”. CRM data was pre-joined in order to answer questions like “show me the aggregate throughput on all inter- faces at this customer account”.

From business impact standpoint, answering these sorts of questions with Glassbeam was, by design, a matter of a few seconds at most. By contrast, the in-house solution was incapable of answering these sorts of questions without

custom-written scripts, which would run for days. One infamous report, which we were able to duplicate and run in a few seconds, had been taking all day to run using the in-house solution, and impacting usability of other tools.

Conclusion

The product intelligence that can be gained from analyzing semi-structured and unstructured machine data offers a complete picture of customer status, configuration changes, and usage. Such intelligence is key to lower costs, higher revenues, and improved customer satisfaction.

While companies can choose to build such an analytical solution in house, it is not worth the time and effort to do so. Smart engineers can build many applications and products, but is that core to your business? Further, creating a DSL such as Glassbeam's Semiotic Parsing Language takes many years and is critical to enterprise-wide deployment and scalability.

Business intelligence projects in the 1990s required cleaning data before it was moved to data warehouses, and much of this work was done in house using custom scripts. Then the category of Extract, Transform, Load (ETL) tools and visual BI were born. The machine log analysis situation is similar, and Glassbeam provides a high-powered next-generation Extract, Load, and Transform (ELT) as well as a scalable platform and datastore with pre-defined search and analytics applications.

Thus, Glassbeam dramatically reduces the total cost of ownership (TCO) by packaging all the required tools for machine data analysis, including SPL and related highly scalable parsing, loading, and analyzing components. Most importantly, with Glassbeam you can gain competitive advantage in a matter of only days or weeks using customer intelligence gleaned from your machine data.

Contact us at sales@glassbeam.com

Glassbeam, Inc.
5201 Great America Parkway, Suite 360 • Santa Clara, CA 95054
Phone: 408-740-4600 • www.glassbeam.com