The Right Tool for the Job: Bringing Together Structured, Unstructured, and Semi-Structured Data to Differentiate Your Business

An Overview of the Vertica/Hadoop Solution
Executive Summary

The explosion of data volume, coupled with the emergence of unstructured and semi-structured data sources, is changing the landscape of business intelligence. Over the past five years, companies have increasingly been looking for a “magic bullet” solution to monetize 100% of their data. Hadoop is only partially equipped to handle this transition. This white paper will explain why to deliver the best results, companies need look no further than an integrated Vertica/Hadoop solution.

Today’s Challenges: Where’s the Data?

“The data warehouse has a few terabytes today, but we have ten times that floating around as unstructured and semi-structured data.”

“We’d love to load all our semi-structured data into the data warehouse, but it’s just not economical to do it.”

“With our current architecture, analyzing the log files from our new application would take a week. But we need it in an hour.”

At Vertica, we hear statements like these almost every day. These three statements capture some of the biggest challenges that analytics teams face in today’s business environment.

First, the nature of data has changed. Merrill Lynch estimates that as much as 85% of all business information exists as unstructured or semi-structured data. In other words, for every terabyte of data analyzed, there are almost six terabytes of data – with new insights – ignored. Agility demands unstructured data. To compete effectively, a social gaming company needs to deploy new in-game features multiple times per day. This means the entire architecture must provide agility, including the ability to log activity. New features will be added and deleted frequently. The entire analytic infrastructure must be agile enough not only to accommodate volume, which can change rapidly, but to accommodate new data streaming in with no advance notice. And as if that weren’t enough, the number of devices capable of generating data streams – smartphones, embedded devices, network equipment, human facing applications, etc. – is exploding. For the purpose of this paper, we define unstructured data as text with very little structure, such as this document, and semi-structured data as data with an identifiable structure, but it may not be rigidly enforced. A classic example of semi-structured data is web log data; there are identifiable fields, but the number of fields and their usage may vary from row to row, and fields may be overloaded with multiple values.

Second, today’s data warehouse solutions were largely designed at a time when structured data was the norm. Built with an appliance mindset, often designed at a time when the mainframe was fresh in the minds of technologists, traditional solutions are
intended to handle a predictable, finite volume of data. Once that capacity is reached, the stepwise cost to add capacity is extremely high, both in terms of direct costs such as hardware and software, and also in terms of indirect costs such as lengthy implementation cycles and downtime. Compounding this problem is that enterprises now need to analyze large volumes of new data before its business value is proven. This requires a data warehouse platform where the economics make it cost-effective to load and analyze this data in order to evaluate its value. The high upgrade costs of many of the existing solutions make this impractical.

Third, the competitive pressure to make a business decision has compressed the window of time to prepare and analyze the data. There was a time, not long ago, when it took a month for reports to reflect new data about a business. Today, social gaming companies routinely demand the ability to deploy a new game feature in a live production setting, and see the results in a dashboard within ten minutes, based on queries against multi-terabyte data warehouses. This demands a rethinking of the pipeline to deliver data. It must have the capability to process extremely large volumes of unstructured data and find the structure, and it must have the ability to provide highly performant query performance with a toolset that doesn’t require exotic skills, so analysts can focus on the analysis, not how it is done.

We can distill this down to an abbreviated list of fundamental shifts taking place:

1. Solutions require increasing amounts of un- and semi-structured data to be effective. And analyzing this data requires analytic engines suited to the task.

2. Data pipelines – from source system to end-user – must be agile enough to handle frequent, unpredictable changes.

3. The volume of data – of all types – is exploding. Systems must be able to scale quickly and without downtime, with economics that favor the inclusion of data whose value is yet to be proven.

**Current Approaches**

The net effect of these shifts is to stress most existing infrastructures to the breaking point. Technology teams will extend and maintain as long as they can, but at some point this is no longer economically feasible. In addition, the increasingly long decision cycles forced by using an aging architecture will put the business at a competitive disadvantage; even if you’re not learning from your data, you can bet your competitors are learning from theirs. And if you take a week to react to new information, one of your competitors will react in a day and compete more effectively.
What we at Vertica typically encounter is some flavor of the architecture depicted in Figure 1. Data warehouse teams know that they need to analyze a broader array of data faster than ever before. They typically begin by extending the current architecture—an ETL infrastructure and row-oriented data warehouse—with complex batch processes and procedural logic. In the early stages, this often works. But it doesn’t scale. The engineering needs to grow such an infrastructure make it cost-prohibitive to extend, and current solutions for ETL and row-oriented data warehousing either do not scale at all due to fundamental architectural limits, or they require such a large investment to grow that it isn’t economically feasible to admit new projects.

Hence, technology teams either start to say “no” (a non-starter for those who wish to keep their jobs), or, more likely, they start to evaluate tools suited to unstructured data such as Hadoop. Very well suited to low initial cost exploratory analysis of unstructured data, Hadoop is increasingly included in the tool portfolio of data warehouse teams. Today, however, it is often deployed in a one-off setting, with integration points that bolt it onto the existing data pipeline in inefficient ways. What we often hear from customers is that they’ve deployed Hadoop in a lab setting for applying structure to semi-structured data such as weblogs, and that the data flow is circular: it draws some data from the existing data warehouse, merges it with web log data, and feeds the results back into the existing ETL processes, which then load to the data warehouse.

As an incremental step towards an enhanced analytic infrastructure, it works. Yet, the multiple integration points result in a fragile flow of data; adding new flows is often time-consuming and costly; and analytic cycles (the time it takes to stream from an incoming flow of raw data to an effective decision) are longer than they should be. When this solution is put into production, the limitations of open-source software make themselves very clear in a hard to maintain infrastructure.

**Figure 1: Common DBMS/Hadoop BI Environment**
Is Hadoop the Silver Bullet?

This is another question we encounter daily. Technology teams, frustrated with the high costs and limitations of traditional row-oriented data warehouses and ETL infrastructures, discover Hadoop and its ecosystem and see it as a potential replacement for their entire data warehouse infrastructure. Like many open source solutions, Hadoop has what appears at first glance to be a solution for every problem. But these are generally non-standard tools with limited feature sets, and they inherit the fundamental limits of Hadoop: a document-oriented, batch-oriented system inconsistently maintained by the open source community. Here’s a quick overview of the Hadoop ecosystem detailing both its strengths and weaknesses.

**Hadoop:** a collection of open source projects which provide a framework for distributed MapReduce computing.

Strengths: scalable, fault-tolerant framework running custom or arbitrary code over a large number of documents in a massively parallel environment. Relatively easy to begin to use on large-scale problems quickly.

Weaknesses: document-oriented; developer-intensive; batch-oriented; lack of manageability features; less performant than SQL databases (due to lack of resource management); lack of cost-based execution optimization.

**Hive:** tool developed at Facebook which allows for a SQL-like language (HQL) access to Hadoop.

Strengths: puts a SQL-like interface in front of Hadoop.

Weaknesses: Huge gaps in SQL compliance; batch-oriented (HQL is compiled to MapReduce, then executed; relies on Hadoop infrastructure with its lack of resource management and execution optimization; query runtimes highly unpredictable.

**Pig:** a framework providing a higher-level language to increase developer productivity in Hadoop.

Strengths: Useful for developers without experience in Java.

Weaknesses: Little more than an abstraction front-end for authoring Hadoop MapReduce jobs.

**HBase:** an open source, column-oriented database modeled after Google’s BigTable.

Strengths: Provides random-access to HDFS data, independent of batch MapReduce.
Weaknesses: Reliant on Hadoop infrastructure (and its corresponding weaknesses); batch-oriented, often resulting in high query latency; job runtimes can be highly variable; lightweight query language; lack of secondary indexing; non-standard query interface.

**Mahout: a project for distributed machine learning (DATA MINING) on Hadoop such as a collaborative filter algorithm for recommendation engines (products you may like)**

Strengths: good set of algorithms available; open source transparency.

Weaknesses: inconsistent implementation of algorithms and inconsistent quality of output; inherits Hadoop’s limitations; developer-intensive; unreliable scalability with large data volumes.

**MongoDB: open source document-oriented database.**

Strengths: strong with semi- or unstructured data; distributed architecture which scales.

Weaknesses: single document transactions only; poorly suited to batch ETL; custom query language based on object-matching.

After reviewing this list, it seems that a Hadoop-only solution will not meet the needs of a business as more than a short-term, interim solution. The tools are built on the foundation of Hadoop and HDFS, and they inherit its limits: lack of resource management and cost-based execution optimization. They are typically a narrow solution to a specific problem; they work for a very narrow set of needs; and the fundamental needs of a Hadoop ecosystem include a team of developers specializing in massively parallel development. Perhaps most importantly, none of these solutions solves the problem of real-time, ad hoc access to data for the analyst – someone who specializes in understanding the data, not the code. That said, Hadoop serves an important role in the analytic decision cycle.

There are problems which Hadoop can solve relatively easily, for which Vertica doesn’t yet have a solution. For example, Vertica doesn’t offer a solution to natural language parsing. This is a problem solvable in Hadoop. Other current Vertica limits, such as pattern matching or classification, can be solved in Hadoop today, and will be addressed in Vertica in the near-term.

**Analytics Cycles: Act on Today’s Information Today**

This is a good time to explain what we mean when we refer to the “analytics cycle” of a business. Like any organism, a business takes in information and responds. The process of taking in raw data and giving it meaning is why data warehouses and business intelligence tools exist. While Curt Monash has detailed six potential use
cases for investigative analytics (http://www.dbms2.com/2011/01/03/the-six-useful-things-you-can-do-with-analytic-technology/), for the purpose of this paper, when we use the term “analytics cycle”, we’re referring to two very common business information/decision flows:

1) **Ad hoc analysis and model formulation**: the collection of a wide array of data, preparation of that data, and application of open-ended, ad hoc analyses by specialists who focus on creating models for decisions. This analytic cycle is highly iterative, with requirements which vary from analysis to analysis, and which can require a large volume of data. Imagine a growing business planning to open a call center. It does not know what information is important to cross- and up-sell for calling customers until it has thoroughly analyzed a wide array of data sources – a highly iterative and unstructured process.

2) **Decision engines**: engines which deploy these decision models in real-time or near-real time settings. This analytic cycle is very much like an application deployment: highly structured, with predictable analysis, and can operate on any volume of data. Imagine a call center with many operators. When it receives a new call, a query is issued against a decision engine to identify the demographics of the caller. The query is always the same, creating a highly predictable system. The power of this engine is in the model; the more accurate it is, the more effective it will be.

The ad hoc analysis cycle is undoubtedly the more important of the two. A predictive model which fails to predict accurately can be worse than having no model at all. In addition, decision models change frequently as conditions change. Consequently, an ad hoc analysis cycle must be both fast and repeatable. There are several fundamentally important enablers to an effective ad hoc analysis cycle:

1) **Rich data**. The richer the data, the more effective the model.

2) **Full volume data**. Sampling adds overhead to the analysis time and introduces the risk of an inaccurate model.

3) **Easy to use**. Analysts specialize in analyzing the data. It is impractical to expect them to write MapReduce Java code or master massively parallel computing strategies. They need tools they can use immediately to focus on the meaning of the data, not the tools used to obtain it.

4) **Real time**. Making a decision quickly requires the ability to query the data even more quickly. Because analysts rarely know the question when they begin analyzing data, they frequently have to run multiple iterations. Minimizing
analysis turnaround time requires the ability to ask questions of, and receive answers from, a data warehouse in real time.

How can a technology team deliver all these capabilities, while coping with the data complexity and flow issues, and effectively include unstructured data?

Hadoop and Vertica should each be used to their strengths: Hadoop for finding structure in loosely structured data, or for applying specialized machine learning models; and Vertica for simple, real-time access to the data by analysts. Rather than allowing circular data flows and multiple redundant methods of gaining visibility into the data, both of which increase the risk of incorrect decisions, we are seeing rationalized flows in which Hadoop is used to apply structure, and Vertica is used to inquire of the newly structured data.

This meets the changing needs of the enterprise as follows:

1) It will cope with both structured and unstructured data in a highly performant fashion. With Hadoop, a schema can be applied to unstructured data, preparing it for analysis in Vertica. Semi-structured data can be loaded to Vertica very quickly and analyzed or transformed to suit analytic needs with a widely understood language. Also, it is highly performant; Vertica’s columnar architecture is uniquely well suited to providing very rapid response times to queries on multi-terabyte tables.

2) It is capable of extremely high rates of throughput so that data can be made available for analysis very quickly, on the order of minutes or seconds.

3) The architecture copes with data changes very easily, both due to feature richness throughout the architecture as well as high performance data manipulation capabilities in ETL, Vertica and Hadoop.

4) Each part is scalable independently of the other. Hadoop and Vertica are very easily scaled with standardized hardware.

5) Each component of the architecture has a rich set of features well suited to the role it plays. Hadoop is a rich MapReduce environment with a variety of programmer-friendly tools for development. Vertica is used with ANSI SQL and supports a large ecosystem of business intelligence and ETL tools. Also, with a large and growing library of analytic functions, Vertica provides a sophisticated toolbox for the analyst.

Existing structured data is easily managed through conventional ETL processes, while unstructured data can be made meaningful with Hadoop. With Hadoop, it is possible to
create a scalable infrastructure and analysis framework and confront the need to apply a schema to unstructured data and prepare it for analysis. This is where we frequently see it being deployed today. Its parallel, batch-oriented processing model makes it well suited to large-scale MapReduce problems such as text analysis.

But sometimes, companies simply need to obtain fast counts of key metrics. Authoring a Hadoop job for this is overly complex. Semi-structured data can easily be loaded into Vertica at high speed and analyzed quickly.

For the analyst, and for a high speed real-time decision engine, Vertica is very effective. Analysts can be more productive with a language they know – SQL – and can analyze and manipulate full volume data at high speed. This can be scaled on an economical infrastructure, with minimal downtime. Vertica is capable of providing response times measured in seconds or milliseconds on terabyte size tables, so it is well suited to large scale decision applications. Finally, it includes a high-speed adapter for moving data into, and out of, Hadoop.

**Conclusion**

To cope with the increasingly rich array of data sources, constant change, rapidly growing volume of data, and complex set of analytics needs of the enterprise, information technology teams will need to evolve their infrastructure to cope in a way that doesn’t break the budget. By using the right tool for the job, this can be achieved. In this paper we’ve discussed one potential architecture which – based on many conversations with current and potential customers – is a very robust solution to the challenges.