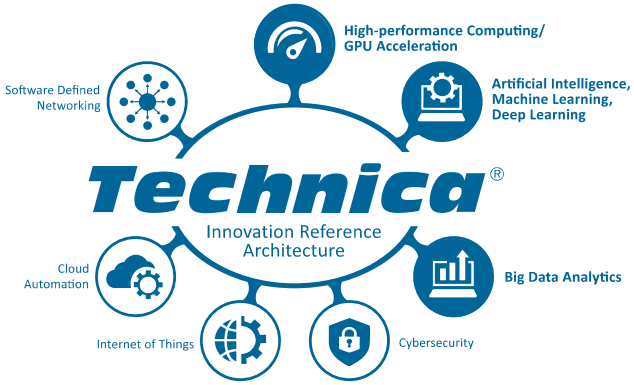
Technica®

Today's enterprise is producing and consuming more data than ever before. Enterprise data storage and processing architectures have struggled to keep up with this exponentially increasing volume, variety, and velocity of data. With the growth of the Internet of Things (IoT) this data will increase by orders of magnitude.

The Technica Innovation Platform White Paper Series presents advanced topics that will drive competitive advantage for next-generation IT over the next three-to-five years.



GPU ACCELERATED BIG DATA ARCHITECTURE

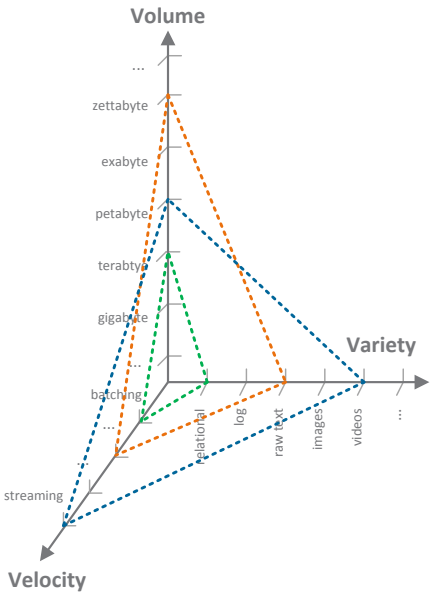


Figure 1 – Three V Characteristics of Big Data

Data volume has increased from the terabyte-range to petabytes and beyond. Data variety has morphed from structured to semi-structured and unstructured pictures, log data, and raw text. And where insight could be derived in the past by batching data, the enterprise has increasingly become real-time. This has necessitated architectures that incorporate real-time, streaming data.

Most importantly, consistently finding actionable insight from this data has become ever more problematic. In response to these dilemmas, enterprise architects have created more complex Big Data architectures. This paper examines various iterations of Big Data architectures as it highlights a very promising next-generation advancement for Big Data analytics—GPU acceleration.

FIRST AND SECOND GENERATION BATCH PROCESSING

The first type of data requiring storage and processing by the enterprise was structured data. SQL databases and data warehouses were implemented widely throughout companies. In the mid-2000's Hadoop was introduced to process less structured data.

Data Warehouse

The concept of the data warehouse originated almost 30 years ago for enterprises with many systems operating in independent silos. IT management replaced inefficient, specially-purposed decision-support systems with a more streamlined model, where business decisions were not hard-wired into software. This approach, whether in the form of a full data warehouse or a more limited data mart, has been the norm.

Data from disparate data stores was Extracted, Transformed, and Loaded (ETL) into the enterprise data warehouse. Once stored, the data could be

analyzed to provide insights and generate reports. This pattern—storing historical data to be analyzed at a later date—is the essence of batch processing. Still deployed widely, the enterprise data warehouse excels at storing and processing structured data, and is effective in analyzing historical operational data for decision making.

NoSQL and Hadoop

With the advent of Internet powerhouses like Google, Facebok, and other social media providers, data warehouses struggled to keep up with the growing velocity, variety, and volume of data. This led to the advent of both the NoSQL movement and Hadoop.

NoSQL databases allow the storage and retrieval of information in models that differ from tabular relations used in relational databases accessed by SQL, i.e. polyglot persistence.

Various types of NoSQL databases exist—each with strengths for specific use cases:

- **Key-Value Store** – Simple hash table keys and values (Riak, Amazon S3)
- **Document** – Stores documents which can be XML, JSON, BSON, etc. (MongoDB, CouchDB)
- **Secure Column-based** – Each storage block contains data from only one column (HBase, Cassandra, Amazon SimpleDB, SQuadron)
- **Graph-based** – Utilizes edges and nodes to represent and store data (Neo4J)

In 2004, Google introduced the MapReduce framework as a simple and powerful programming model that enables the easy development of scalable parallel applications to process vast amounts of data on large clusters of commodity machines. The framework ultimately merged into the Hadoop open source project.

The key elements at the core of Hadoop were MapReduce married to a NoSQL data store—typically HBase—spread out over a cluster of machines. By allowing storage of a variety of datatypes (structured, semi-structured, unstructured) in a NoSQL database and the distributed processing of data with MapReduce, Hadoop overcame many limitations of the traditional warehouse. Over time, a vast number of specialized open source projects developed into a robust Hadoop ecosystem.

While both data warehouses and Hadoop excelled with the batch processing of data, neither was designed to work with real-time data. Real-time data can certainly be batch processed; however, the “right-now” aspect of the insights are lost. In response, recent advances in Big Data architectures have sought to incorporate real-time, streaming data.

STREAM PROCESSING WITH LAMBDA ARCHITECTURE

Computing arbitrary functions on a dataset in real-time is a daunting problem. There is no single tool that provides a complete solution. Instead, a variety of tools and techniques must be utilized to build a complete Big Data system. Collectively, this solution is known as the Lambda Architecture.

Lambda Architecture Explained

In response to the need to process streaming data, the Lambda Architecture was proposed in 2013. To reduce the complexity seen in real-time analytics pipelines, the Lambda Architecture constrains incremental computations to a small portion of the architecture. The result is a scalable real-time system that is able to deal with the “bursty” nature of streaming data.

The Lambda Architecture consists of two specially tailored layers:

- **Speed Layer** – A "hot" path for latency-sensitive data flows where results need to be ready in seconds or less. Analytics clients can use the data for both machine and human consumption.
- **Batch Layer** – A "cold" path where data is processed in batches, and latencies of minutes or even hours can be tolerated.

With the Lambda Architecture, there are two paths for data to flow in the pipeline (See **Figure 2**).

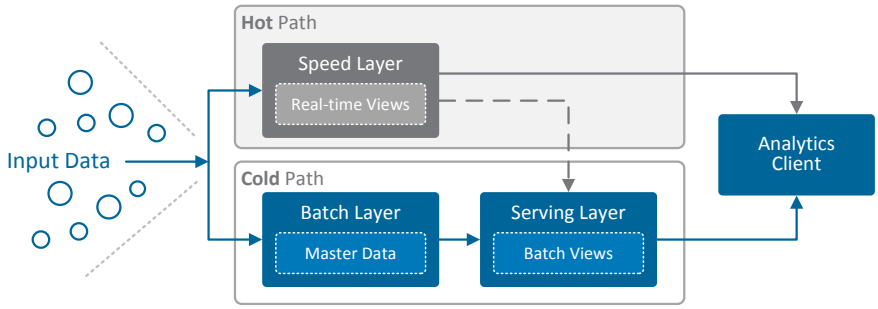


Figure 2 – Notional Lambda Architecture

All data entering the pipeline flows into the Batch Layer—the cold path. This data is immutable and labeled Master Data in Figure 2. Any modifications to the value of particular datum are reflected by a new, time-stamped datum being stored in the system alongside any previous values. This approach enables the system to re-compute the then-current value of a particular piece of information for any point in time across the history of the data collected. Because the cold path can tolerate greater latency, the computation can afford to run across large data sets, and the calculations performed can be time-intensive. The objective of the cold path is to achieve extremely accurate results. Data processed by the Batch layer is output to a Serving Layer in the form of Batch Views.

When data moves into the Speed Layer—the hot path—the data is mutable and can be updated in place. As the results are typically desired in near-real-time, the speed layer places a latency constraint on the data to limit the types of calculations that can be performed. This might mean switching from an algorithm that provides perfect accuracy, to one that provides an approximation. For example, instead of counting the unique number of visitors to a highly-trafficked website, an educated guess may be calculated using an algorithm like HyperLogLog. The objective of the hot path is to trade off some amount of accuracy in the results to ensure that the data is ready as quickly as possible. Latency sensitive calculations are applied to the input data by the Speed Layer and exposed as Real-time Views.

The Speed Layer and the Batch Layer converge at the Analytics Client application. The client must choose the path from which it acquires the result: either the less accurate but most up-to-date result from the hot path, or the less timely but more accurate result from the cold path. An important component of this decision relates to the window of time for which only the hot path has a result, as the cold path has not yet computed the result; i.e. the batch process has not completed. The results from the Speed Layer are only active for a small window of time. Eventually these results will be updated by the more accurate batch layer. As a result, the volume of data the hot path must process is minimized.

The motivation for the creation of the Lambda Architecture may be less obvious than first apparent. It is true that enabling a simpler architecture for real-time data processing is important. However, the primary reason for the Architecture is

to provide human fault tolerance. In effect, the Lambda Architecture recognizes that we can actually maintain all of the raw data. At the same time, it recognizes that bugs happen, even in production. Lambda Architectures offer a solution that is not just resilient to system failure, but tolerant of human mistakes because it maintains all the input data and has the ability to re-compute any errant calculation through batch computation in the cold path.

Because a Lambda architected system will be able to handle much larger amounts of data, more data can be collected, thereby providing more opportunities for valuable insights. Increasing the amount and types of data stored will provide greater prospects for data mining, analytics, and ultimately the ability to create a greater range of applications. This last point is key: the Lambda Architecture enables greater range of analytics applications and the adoption of new analytics methodologies like GPU acceleration. Before moving to a detailed discussion of GPU acceleration, we will look at an implementation of the Lambda Architecture proposed by Technica.

Sample Big Data System Incorporating Lambda Architecture

The solution pictured in **Figure 3**, dubbed "Sentry", was created by Technica in response to a proposal for an intelligence system that monitors social media in real-time.

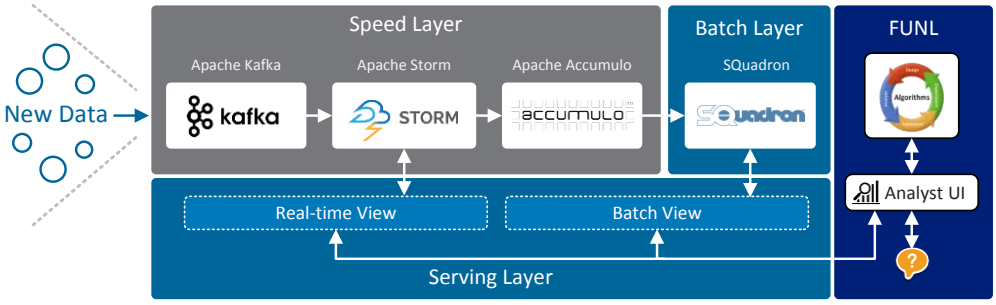


Figure 3 – Sample Lambda Architecture of Sentry

Sentry is composed of open source software from the Apache Software Foundation, augmented with Technica's FUNL and SQuadron solutions. Technica's Independent Research and Development (IR&D) organization has spent four years in the development of FUNL and SQuadron.

As portrayed in the figure, new data sources are integrated using Apache Kafka's publish/subscribe mechanism. Apache Storm is a free and open source distributed real-time computation system that facilitates the reliable processing of unbounded streams of data.

Kafka and Storm are essential elements of Sentry's Speed Layer, and fairly standard in Lambda architected systems. Alternatively, many enterprises select Apache Spark instead of Storm. Data in the Speed Layer is persisted into Apache Accumulo, which has robust security features, including cell level security.

Data from Accumulo is extracted, transformed, and loaded into SQuadron, which serves as a GPU-accelerated data warehouse. GPU acceleration will be described in following sections. The ETL process can be used to scrub data and apply anonymization techniques as needed.

SQuadron is a GPU accelerated database. It was developed and tuned to take advantage of the multi-core nature of GPUs to provide fast response time to analytic queries on large datasets. The combination of a column-based storage engine, fast compression and decompression, querying on compressed data, and other design enhancements provide exceptional query performance on commodity hardware systems with a small power consumption footprint.

FUNL is a GPU accelerated graph and social media analytics engine. It enables management, analysts, and data scientists to utilize graph analytic algorithms and Artificial Intelligence (AI)—machine learning and deep learning algorithms— to identify meaning from large-scale data. Anonymization strategies can be employed within the FUNL algorithms to protect privacy, civil rights, and civil liberties.

GPUs

Technica's Sentry implements a GPU accelerated Big Data architecture following Lambda principles. Other papers in this series provide more information on the details of GPU acceleration. In short, GPU acceleration splits processing tasks between multi-core CPUs and multi-core GPUs—GPUs just have 1000s more cores.

Harnessing the power of GPUs for Big Data is a relatively recent phenomenon. In the past, GPUs were largely associated with display hardware, video gaming, or high-priced supercomputers. Increasingly the massively parallel capabilities of multi-core GPUs are being exploited to radically speed up data mining, machine learning, and deep learning algorithms—often by a factor of 100X—at a fraction of the cost of comparably priced CPU-only hardware.

In fact, deep learning is one of the hottest area of machine learning. Deep learning uses deep neural networks to teach computers to recognize patterns. To date, its greatest success has been in image and speech recognition. However, numerous other use cases for deep learning exist. Technica, for example, is using deep neural networks to detect anomalies in network traffic.

GPU ACCELERATED ARCHITECTURE

We have briefly highlighted the progression in enterprise architecture in an effort to process the volume, velocity, and variety of expanding data, i.e. the evolution from SQL databases/ data warehouses, to Hadoop, to the Lambda Architecture. However, the core Apache technologies of the Lambda Stack will be overwhelmed by the coming zettabytes of the near-future. GPU acceleration will be a requirement for analytics in the next generation Big Data architectures.

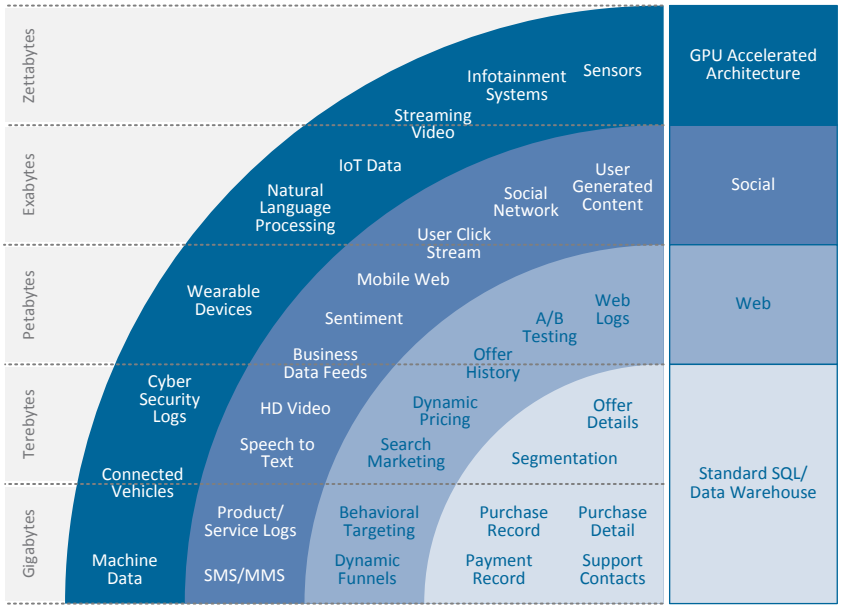


Figure 4 – Big Data Progression in Volume, Variety, and Velocity

This new architecture—the Gamma Architecture—adds GPU accelerated software and algorithms that leverage the parallel processing power of GPUs for appropriate workloads.

Figure 5 presents a representative sample of a Gamma Architecture. While not exhaustive, the figure identifies product alternatives for the various components.

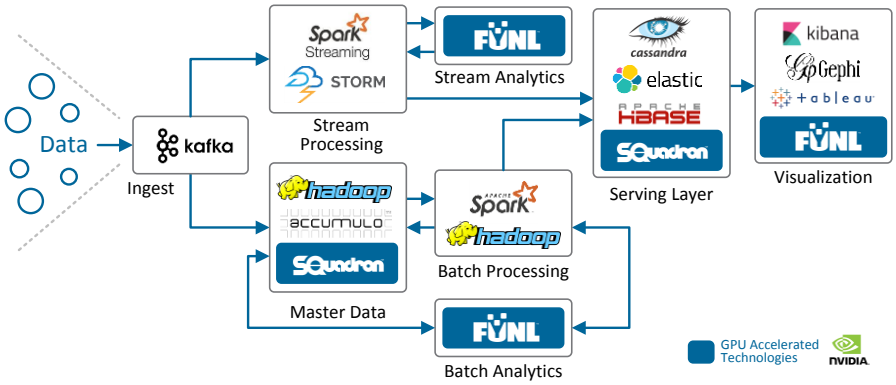


Figure 5 – Sample Gamma Architecture

The Gamma Architecture provides a robust, future-proof platform that can scale with massive data volumes. The Architecture truly excels at enabling next-generation analytics.

Sample technologies of the Gamma Architecture include:

- **NVIDIA GPUs** – Accessible by CUDA language (Technica uses NVIDIA GPUs in FUNL and SQuadron)
- **Apache Cassandra** – Robust NoSQL database
- **Apache HBase** – SQL-compliant database client that typically resides on top of Hadoop File System (HDFS)
- **Elasticsearch** – Open source, free-text search engine. Technica has created a plugin for Elasticsearch called TopicD. TopicD works with FUNL to move beyond enterprise search to perform GPU accelerated enterprise discovery
- **Kibana** – Visualization platform for Elasticsearch
- **Gephi** – Open source product that provides visualization of graphs
- **Tableau** – Business intelligence and analytics visualization software

GPU ACCELERATED ANALTYTICS

The Gamma Architecture enables GPU accelerated analytics. Technica recognized early on that the coming data deluge required new strategies. In 2012, Technica's IR&D organization began experimenting with GPU acceleration utilizing NVIDIA GPUs. The result was FUNL, with GPU-accelerated graph analytic, machine learning, and deep learning algorithms that provide a range of analytic solutions for a wide variety of problems.

Moreover, each element on the analytics progression, as shown in **Figure 6**, is GPU accelerated.

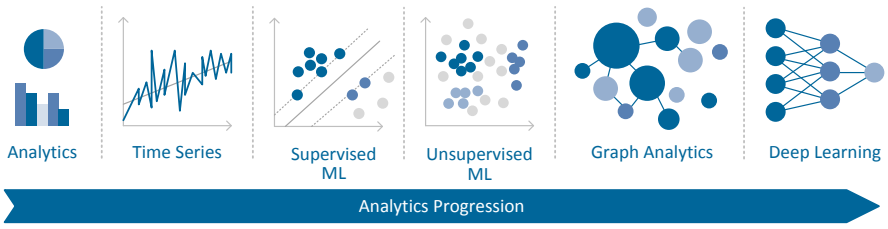


Figure 6 – Progression of Analytics Solutions Enabled by Gamma Architecture

Technica is currently developing a network anomaly detection deep learning algorithm. Technica uses historical network data to train the neural network in a batch process (the batch processing element of the Gamma Architecture). The trained neural network can be deployed to recognize network anomalies, e.g., inappropriate IP addresses, malicious activity, etc., in real-time (the stream analytics component of the Gamma Architecture).

SUMMARY

The movement towards a GPU-accelerated Big Data architecture involved a progression of technologies to process the volume and variety of expanding data.

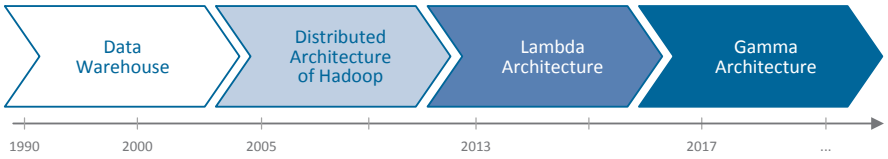


Figure 7 – Progression to a Gamma Architecture

Each technology leveraged the strength of its predecessor, while providing expanded capabilities. The Gamma Architecture provides the enterprise with unprecedented analytical capabilities and near-supercomputer performance – without supercomputer prices.

Technica provides professional services, products, and innovative technology solutions to the Federal Government. We specialize in network operations and infrastructure; cyber defense and security; government application integration; systems engineering and training; and product research, deployment planning, and support.

Technica®
22970 Indian Creek Drive, Suite 500
Dulles, VA 20166
703.662.2000