Streamlining the BigData Landscape: Real World Network Security Usecase

|  |
| --- |
| **Sonali Parthasarathy**Accenture Technology Labssonali.parthasarathy@accenture.com |

## ABSTRACT

In this paper, we introduce the concept of data acceleration and how it addresses the challenges of data movement, processing and interactivity. We propose a taxonomy to categorize the current Big Data technologies based on the challenges they address. We will demonstrate how these architecture categorizations have been applied to a real world use case. Using ArcSight data collected in the form of network security log files from a large North American network, we detect anomalous events and present the results using interactive visualizations. We describe our security use case technology stack which consists of data exploration tools (Tableau), real-time event processing and analytics tools (Kafka, Apama, GemFireXD, CDH5) and interactive visualizations (D3.js). Finally we will simplify the architecture decision-making process by prescribing the common architecture patterns.

# AUDIENCE

[Big Data Architecture],[In-Memory Technologies],[Data Visualization], [Real-Time event processing], [Analytics]

[Intermediate Talk]

1. **INTRODUCTION**

A modern data supply chain starts with data being created. As the data moves through the links of the chain, it incrementally accrues vales ending in actionable business insights such as ideas for a new product, process innovations or global strategies.

In this presentation we introduce the concept of data acceleration which plays a major roles in the data supply chain. “In its simplest form, data acceleration stems from tools and techniques that enable massive amounts of data to be ingested, stored, and accessed at lightning speeds”[[1]](#endnote-1). Data acceleration helps organizations address three challenges: Movement; how to move data swiftly from its source to places in the organization where it is needed, Processing; how to process it to gain actionable insights as quickly possible, and Interactivity; how to foster faster responses to queries submitted by users or applications.

We identified six different data technology components to build the architecture needed to support data acceleration which are described in section II. Each component can address data movement, processing, and/or interactivity, and each has distinctive technology features that help address these challenges.

1. **CLASSIFICATION**

**Big Data Platform (BDP):** This construct is a distributed file system and compute engine that can be used to facilitate data movement and processing. BDPs contain what we call a Big Data Core (BDC)—a computer cluster with distributed data storage and computing power. Advancements in big data technologies have enabled BDCs to function as a platform for additional types of computing, some of which (like query engines) can specifically support data interactivity.

**Ingestion:** This construct is used for collecting, capturing, and moving data from its sources to underlying repositories where users can process it. Traditional ingestion was done in an extract-transform-load (ETL) method aimed at structured data for batch loads. Modern techniques involve ingesting streaming data from disparate sources and getting them to an appropriate landing place.

**Complex Event Processing:** This construct is a method of tracking and processing streams of data about events (such as click streams or video feeds) and triggering an action immediately based on detected patterns to deliver actionable insights to the decision makers. A quick example is validating security events against previously verified breaches of information in real time to assess new threats.

**In-Memory Database:** It is a database management system that relies primarily on main memory for computer data storage. It differs from traditional database management systems that use a disk storage mechanism. Accessing data in memory eliminates the “seek time” involved in querying data on disk storage, thus providing speedier and more predictable performance.

**Cache Cluster:** It is a clusters of servers in which memory is managed by central software and can offer high-speed access to frequently accessed data. The clusters are used when there is an extremely high volume of reads from multiple sources of data that does not change often, or when a database is stored on disk where seek time can be sub-optimal.

**Appliance:** It is aprepackaged or preconfigured set of hardware (servers, memory, storage, and I/O channels), software (operating system, database management system, and administrative management software), and support services. It is sold as a unit, typically with built-in hardware redundancy, which helps confirm that the device will remain available in the event of a component failure.

1. **ARCHITECTURE PATTERNS**

The architecture components described above cannot function in isolation to support data acceleration. Instead, they must “play well” with each other, capitalizing on one another’s advantages. In the table below, we see four fundamental technology stacks that meet these imperatives. We use an incremental, add-on approach to show how these stacks are built to enable data movement, processing, and interactivity. Each pattern is classified as Basic, Enhanced or Enhanced+ based on the challenges they address (Fig 1).

|  |  |  |  |
| --- | --- | --- | --- |
| Pattern | Movement | Processing | Interactivity |
| Appliance Only | Enhanced | Enhanced | Enhanced |
| BDP to Appliance | Enhanced | Enhanced | Enhanced |
| Streaming to Appliance | Enhanced | Enhanced+ | Enhanced |
| BDP Only | Enhanced | Basic | Basic |
| Streaming to BDP | Enhanced+ | Enhanced | Basic |
| BDP with In-Memory Analytics | Enhanced | Enhanced | Enhanced |
| Streaming to BDP w/ In-Mem Analytics | Enhanced+ | Enhanced+ | Enhanced |
| BDP with query engine | Enhanced | Basic | Enhanced |
| Streaming to BDP with query engine | Enhanced+ | Enhanced+ | Enhanced |
| Distributed Cache Cluster Only | Enhanced | Basic | Enhanced |
| BDP to cache cluster | Enhanced | Basic | Enhanced |
| In-memory database Only | Enhanced | Basic | Enhanced |
| BDP to In-memory Database | Enhanced | Basic | Enhanced |
| Streaming to In-Memory Database | Enhanced+ | Enhanced+ | Enhanced |

Figure 1: Architecture Patterns

The key takeaways from the above table include:

* Complex event processing can enhance streaming ingestion
* Complex event processing can increase speed by pre-processing data
* Caches and In-Memory databases can enable real-time interactivity.

The solution landscape when depicted architecturally looks like this:



Figure 2: Solution Landscape

1. **NETWORK SECURITY USECASE**

Our use case involved ingesting large amounts of network security events, detecting risky, anomalous events in real-time based on an analytical model, storing the processed data in an in-memory database and creating custom, interactive visualizations to derive insights for the end user i.e. Chief Incidence Security Officer(CISO). The following sections describe the steps involved and the technologies used.

**Data Discovery and Exploration**

Our dataset for this use case was network security events collected from ArcSight. ArcSight is a Security Information and Event Management system which collects log files of network security events from all the devices on your network such as laptops, workstations, servers, routers, switches and intrusion detection systems and anti-virus systems. We received a nightly feed of encrypted, compressed logs from ArcSight into our staging server. Each file consisted of 10 minute chunks of data amounting to 4GB uncompressed. Each record in the log file consisted of 466 fields used to organize and classify each of the events. After doing some initial aggregation of data to understand the distribution, we used Tableau to visually explore the data. We created different types of visualization such as word clouds and bubble charts to understand where most of the events are generated.

**Analytics**

We used the Log Content Analytics (LCA)[[2]](#endnote-2), an internal tool built by Accenture Technology Labs to uniquely identify events, event probabilities and statistics, and discover the temporal relationships between events. We applied the LCA framework to 3 months (2 PB) of security data to generate a graph with nodes representing the events, edges connecting events that are related to each other, the size representing the anomalousness and the color representing the probability of occurrence of the events (Fig 3).



Figure 3: Log content Analytics Graph Explorer

We then used the Anomalous Behavior Analysis Tool (ABAT) module of the LCA framework which uses k-means clustering on the ranked anomalousness values to produce 10 distinct groups ranking scores from high anomalousness to low anomalousness with a randomly seeded centroid and a defined centroid displacement value indicating stability. These are the values that will be fed into the real-time complex event processing engine to detect the anomalous/risky events. This framework was built on Cloudera Hadoop (CDH5) running python MapReduce jobs.

**Real-time processing**

For real-time processing we used a real-time stream simulator script (Java), a messaging queue (Kafka), a complex event processing engine (Apama) and an in-memory database (GemFireXD). The java script creates a simulated stream from the log files sitting on the staging server. From here, each line of the file (record) is read and written to the Kafka messaging queue on a specific topic. The consumer of this topic is Apama which specializes in rapid filtering, aggregating & detecting patterns across multiple streams of real-time data. Specifically we encode the Kafka consumer logic and the results of the ABAT analytics from the previous section (List of anomalous events and their probabilities and rankings) in Apama. When Apama encounters a risky event, it flags it as such and stores the events in GemFireXD. GemFireXD is a real-time, distributed, in-memory, relational database that stores these risky events and their associated attributes with a TTL of 6 hrs.

**Data visualization**

The goal of visualization is making the data accessible to users and applications by connecting to the data infrastructure in universally acceptable ways and ensuring that query results are delivered as quickly as required. We created two visualizations using D3.js, a JavaScript library for building custom visualizations. The first is a streaming visualization which provides the CISO and security incident response teams with board visibility into events happening in different security layers (defense in depth). The teams can watch for fluctuations in the shield to ascertain volume of events at different levels. They can also compare ABAT count with All Event count to understand where anomalous events are occurring (Fig 4).



Figure 4: Security Shield Visualization

The second visualization enables visibility and exploration into the anomalous behaviors of systems on the network. It is a force-directed graph showing the attacker IPs and the target IPs. This allows you to explore the attack patterns and proactively identify outliers to prevent event escalation (Fig 5).



Figure 5: Tactical Network Display Visualization

Putting it all together Figure 6 shows the architecture for our usecase. We use the Big Data platform to do analytical modeling, Ingestion & Complex event processing to detect the risky events at real time, and In-Memory database to store the data and query them real-time for interactive visualizations.



Figure 6: Architecture of the Network Security Use Case

**CONCLUSION**

The outcomes of this papers is two-fold. First and foremost, the paper provides a deeper understanding of the big data landscape by providing a categorization of the technologies based on the key technology challenges encountered when building a technology stack for a given use case. Second, it provides a real world use case to demonstrate how the different pieces of the technology stack can be put together to solve a real world problem.

**REFERENCES/BIBLIOGRAPGHY**

This work is based on a Point of View (PoV) published internally at Accenture. Data Acceleration: Architecture for the Modern Data Supply Chain, 2014

1. Parthasarathy, S., Dukatz, C.M., and Li, Z., Data Acceleration: Architecture for the Modern Data Supply Chain, 2014 [↑](#endnote-ref-1)
2. Puri, C., Accenture Log Content Analytics Platform, 2014 [↑](#endnote-ref-2)