BIDCEP: A VISION OF BIG DATA COMPLEX EVENT PROCESSING FOR NEAR REAL TIME DATA STREAMING

POSITION PAPER – A PRACTITIONER VIEW

Ethan Hadar

Communities Informatics, Zefat Academic College, Zefat, Israel
ethan.hadar@gmail.com

Abstract: This position paper aims to trigger a technical discussion by proposing a conceptual architecture for big data streaming integrated with complex event processing (BiDCEP). BiDCEP expands the Lambda and Kappa (LK) architectures for big data streaming to fit the complex event processing (CEP) and event management domains of enterprise IT. BiDCEP links CEP components as defined in previous work of Events Collections, Purifications and Enrichments with the big data LK batch and speed layers, and wraps the LK service layer with integration interfaces for expandable grid of interlinked BiDCEP units. The BiDCEP architecture can enable the LK big data quality attributes of scale, availability and latency to be maintained, while accounting for CEP enterprise IT requirements of load and content shedding, basic and derived enrichment, semantics transformation, and security enforcement. As such, open source big data streaming strengths can be employed within the context of an enterprise-grade IT with monitored service levels.

Keywords: Data streaming, big data, complex event processing, enterprise integration, event management and automation, precision agriculture.

1 Introduction

Today's IT business stakeholders expect to maximize the usage of their data originated from different sources and settings for different use cases. In particular, they strive to maintain a scalable modular architecture (Gal and Hadar, 2010) for complex event processing (CEP) that combines the benefits of a big data infrastructure for:

- Serving a single data consumer.
- Serving multiple consumers of the same product.
- Creating a cascade of data flow and a grid of data consumers and providers for integrated solutions.

Some of the CEP requirements employed in previous field work (Gal and Hadar, 2010) for enterprise IT grade solutions are captured as components, such as (1) purification component for removing noise and data cleansing and de-duplication; (2) enrichment component that adds meta-data of sampling rate, source of data and time stamp; and (3) semantics expansion component for adding information such as tags or headers for self-explanatory data schema.

Yet, all these CEP components should leverage the advantages of big data streaming architecture such as Lambda and Kappa (LK) architectures (Martz and Warren, 2015), and support low latency events management as well as high latency computed analytics (Hirzel et al, 2013).

This practitioner position paper addresses the combination of the CEP with LK, and proposes a conceptual architecture approach termed BiDCEP (Big Data CEP). The paper triggers a technical discussion on the merits of combining the LK big data streaming architecture with complex event processing architecture, as well as presents a motivational example.

The rest of the paper is organized as follows: The background section reviews the required CEP stages and big data streaming constraints in an enterprise IT organization. Section 3 reviews the requirements and challenges when mixing data streaming, CEP, and big data quality needs. Section 4 details the conceptual architecture and how it links CEP modules of events collections, purifications and enrichments as defined by Gal and Hadar (2010) with big data Lambda and Kappa (LK) architectures batch and speed layers, and how modular CEP should wrap the service layer with integration interfaces for a single consumer or a grid of interlinked BiDCEP units. Section 5 presents a motivational example to trigger a technical discussion of a Precision Agriculture that employs both big data type of information as well as complex event processing. Lastly, a discussion about the potential quality attributes improvements, as well as future usages and opportunities for extensions are presented.

2 Background

Common approaches for data sharing within enterprise integration frameworks may include integrations using the hub-and-spoke canonical model (Reeve, 2013), data warehousing for IT, and ERP for IT such as ITIL (ITIL, 2015), or semantically enriched CEP solution such as Event Management and Automation (Park et al., 2012). Such approaches are based on structured or semi-structured data repositories and XML/JSON-based documents. Each product collects data, purifies, enriches, and aggregates the data, for the usage of reporting dashboards or command and control consoles. Gal and Hadar (2010) proposed a modular structure for CEP units for a single product, as well as a CEP grid for interconnected products to support event management for real time systems. The proposal defined several components and their capabilities, including data collection, purification, storing, enrichment, inferencing, and cross components policy and situation management. The implementation of these components was part of an enterprise IT product set in the domain of Application Performance Monitoring of IT management. In these classical CEP solutions, providing data to consumers and other CEP grid elements is handled after data is preprocessed, stored and structured, as done by the Extract-Transform-Load (ETL) approach for data warehouse solutions (Van der Lans, 2012). This approach fits well with state-of-the-art IT situational applications and UI widgets portals such as dashboards and reporting tools.

Advanced big data analytics scenarios require historical raw data, such as in machine learning (ML) or natural language processing (NLP) techniques. In such cases, additional data that is introduced to the historical dataset can change the computation and system cognitive learning abilities (Kelly, 2015). In addition, when an evolution to the computation algorithm is introduced, re-run of the computation over the entire raw dataset is required. As such, historical raw data persistency system is needed. One of the approaches to resolve the need for both historical and near real time data streaming computation is the contemporary Lambda architecture (Martz and Warren, 2015), which duplicates incoming data into parallel processing lanes at different speeds, primarily for the consumption of a single product. As presented in Figure 1, the Lambda architecture approach is to split the batch and speed processing layers into parallel lanes, repeating the same processing code. Later, within the serving layer, the computation results of the slow yet accurate batch layer are merged with the rapid and good-enough speed layer ones. Thus, a data subscriber can have an improved compromise between the Consistency, Availability, and Partition-tolerance (CAP) elements (Martz, 2011).

The Lambda architecture supports quality criteria such as fault-tolerance, linear scale-out, and extensibility. The batch, speed and serving layers are tuned to handle both high latency computations that use historical data, as well as low latency ones that deal with recent data. This architecture and the specific selection of implementation technology of Hadoop for the batch layer, Impala for the serving layer, and Storm and HBase for the speed layer, cater for a single consuming application. In order to avoid data aggregation code duplication between the batch and speed layers, a variation of the solution, the Kappa architecture, was proposed (Kreps, 2014). The Kappa architecture is focused on a single stream technology that performs historical re-computation according to needs. A batch-processing cluster performs the near real time computation, and is removed from production once completed. Both architectures handle low and high latency computation, and enable re-computation when a data change is detected, or an algorithm change is introduced.

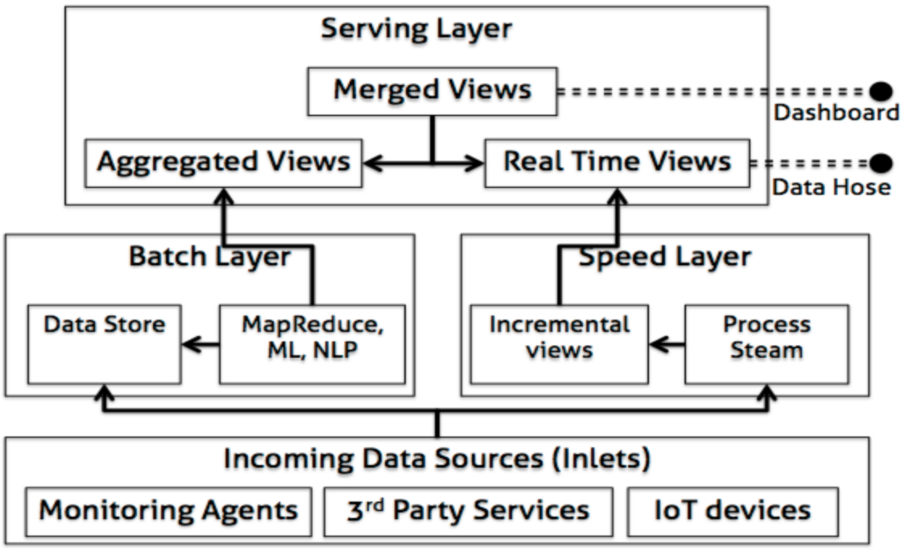

Figure 1: The Lambda architecture for big data streaming of both real time incremental and batch aggregated computations.

3 The Challenge

CEP solutions provide a good solution for reporting tools providing descriptive analytics such as business intelligence, OLAP, and reporting visualizations, as well as automated command and control tools that discover phenomena and examine their sources based on statistics.

Yet, the approach needs to cope with near real-time escalations of big data streaming solutions, where the data creation velocity and volume as well as format transformation are needed as pre-processing stages prior to discovering a new event. The CEP data load and content shedding (Gal and Hadar, 2010) of raw data does not fit well with cognitive computing, machine learning and adaptive solutions such as Predictive Analytics. Such analytics approaches require historical raw data for the learning steps, as well as regressively re-computing the decision-making criteria. CEP engines mostly archive their historical data for presentation needs only.

Serving different stakeholders, data streaming should expose data hoses at different levels of the processing stream (Genkin, 2007)(Hadar and Perreira, 2007), for CEP and analytics problems alike. Both CEP and LK should control data speed throttling, computation algorithms modifications, pluggable visualizations for users interactions, as well as ensure service levels for their quality attributes such as availability, reliability, and recoverability.

Accordingly, a combined BiDCEP conceptual architecture should include:

- Legacy integration value maintenance. The hub-and-spoke and publish/subscribe patterns should be kept for basic enterprise integration needs (Reeve, 2013).
- Open source-based implementation. Architects should be able to select their own internal implementation technologies, such as secured pipe-and-filter (Fernandez and Ortega-Arjona, 2009), big data repository and analytics publishing.
- Data sampling and granularity control (Hadar and Perreira, 2007). Velocity and volume should be tuned according to external configuration policy and store pattern, without changing the architecture (Homer et al, 2014).
- Data enrichment. Data structure variability should be easily transformed to enrich the processed data (Ait-Sadoune and Ait-Ameur, 2010), starting at the agent level, and up to the serving layer.
- Scale according to the read/write nature of the business data usage. Persistency systems should scale using inner big data repository capacities, improving performance and customer-value proposition.
- Modular data resolution and latency response. Both raw data and compound data should be consumed and produced according to business needs (Kinley, 2013).
- Footprint reduction. Sharing of the same measurement agents and tools across products should be employed while maintaining data integrity.

4 The BiDCEP conceptual architecture

Adhering to the challenges detailed in the previous section, the BiDCEP conceptual architecture expends and enhances the Lambda architecture with CEP components as depicted in Figure 2. By using RESTfull services, the inner Lambda implementation components can use Apache Hadoop, Elephant DB and Impala DB for the batch layer, Apache Spark, Flum or Storm for the speed layer, and Druid, Cassandra and HBase for the serving layer. The CEP components can be constructed from elements such as CA Technologies EMA, or IBM Tivoli.

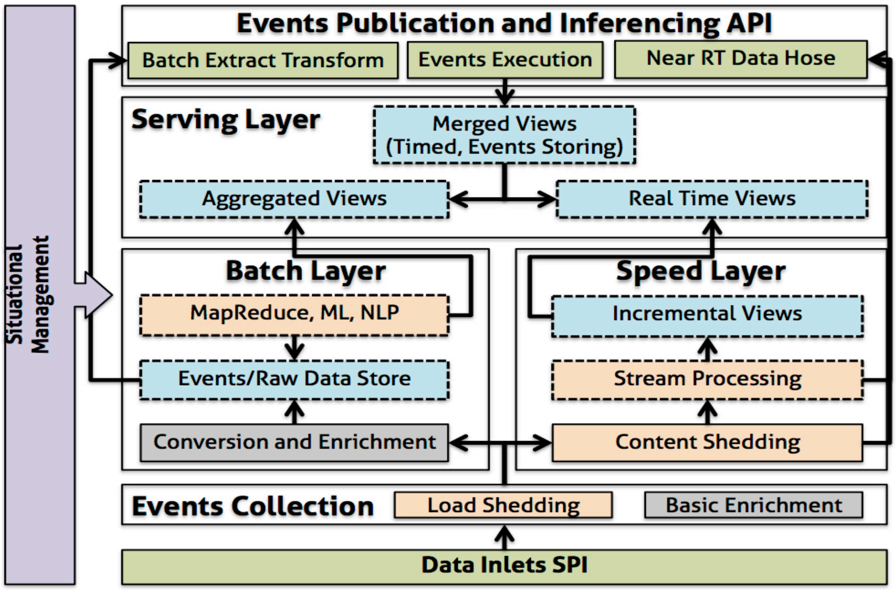

Figure 2: The BiDCEP architecture for interweaved Complex Event Processing (CEP) and Big Data Lambda architecture for data streaming. Dashed blocks are the Lambda components, and solid blocks are the CEP extensions ones.

The main components of the BiDCEP conceptual architecture, depicted in Figure 2 are:

- Data Inlets Server Provided Interface (SPI). This component is responsible for connecting and disconnecting adapters to data sources, encapsulating communication of synchronic and a-synchronic protocols, ensuring payload guaranty delivery, and providing security access authorization.
- Events Collection. This component is responsible for data reduction (load shedding) that is defined by matching computation latency and business usages differences or by adding time delay to avoid jitter.
- Basic Enrichment. This component can enrich measurement with timestamp, data source origin, and other inferred information or meta-data to be used for analytics purposes, appended as batches or per event, according to business needs. This approach can reduce payload size for Internet-of-Things (IoT) devices, delivering only measurements after the protocol initial identification and message envelope data transfer.
- The Lambda Batch layer expanded with Conversion and Enrichment CEP components (Gal and Hadar, 2010). These components are used in cases where the

enrichment processing is computational by nature and slower than simple appending of information, as is usually done in Basic Enrichment component. Examples are data type conversion from strings to integers, and mathematical evaluation of a raw measurement for creating an informative status. This step can be performed after the raw data persistency stage as well. In this case, the enriched information will be stored in the aggregated serving layer.

- The Speed layer enhanced with the Data Content Shedding component. The shedding component is responsible for noise reduction in which data is filtered according to business needs. Filtering rules can remove duplicated information or filter out a subnet. The Content Shedding component requires data transformation, interpretation, logic rules, and decision. As such, deferring actions to the slow batch layer, within the Map Reduce and computation aggregation step, is recommended.
- The Batch Extract-Transform. This component extracts raw yet enriched data from the historical Big Data repository, without the mathematical transformation created by the Map Reduce component or other derived historical computation. Consequently, other BiDCEP units or consumers can use the data without additional manipulation.
- Events Execution. This component is the classical CEP component that drives event actions for command and control or monitoring consumer applications. However, this component wraps the serving layer of the Lambda architecture, in order to segregate consumers' access and their requested data structure, abstraction level, and performance needs.
- Near Real Time Data Hose. This component provides the same ability as the Batch Extract Transform, yet is employed on the near real time content reduced data stream, for those applications that require low latency data hoses. Both the Data Inlets SPI component, and the Event Publication and Inferencing API component, expose and consume the same type of interface so that each BiDCEP unit can serve as a data source to other BiDCEP units, forming a grid of BiDCEPs.
- The Situational Management component. This component is responsible for configuration of the BiDCEP components' including inlets sampling rate; definitions of enrichment; activation of data adapters; CEP rules setting; grid definitions of publishers and subscribers connectivity; authentication and authorization policies; and all other streaming configurations.

The proposed conceptual architecture binds both legacy integration and big data open-source implementations. Velocity and volume can be tuned according to external policy and business needs, and data can be enriched in several steps. Linear scalability can be achieved according to the inherent proprieties of the big data components, and latency properties can be adjusted for both low and high latency consumers according to the LK structure. Overall merging data source adapters technologies of the LK and CEP architectures can reduce footprint, and a modular construction of a data grid for segregation of control can be achieved.

5 Motivational Example: Precision Agriculture

As a triggering example for a technical discussion, consider a smart farm with automated irrigation systems (William, 2015), as an implementation for Precision Agriculture (Precision Agriculture, 2015). In this example, some sensors gather information about the condition of the soil moisture, while others measure humidity correlated with a GIS positioning. Additional data is gathered from weather channels, such as historical forecast and actual conditions, and from social networks. The requested insights should adapt and predict the irrigation and fertilizer automated system, in order to optimize crops' yield with reduced operational costs.

The system compound health encompasses the IT systems and applications side availability as well as the things of the IoT, such as sensors, actuators, cellular connectivity transport, Wi-Fi and cellular grid, mobile control devices, and more. In order to activate the right valve at the right time, the prediction levels and planning requires end-to-end availability of the systems, including monitoring, problem detection, capacity and configuration management. For instance, an irrigation plan should consider the physical condition and flow capacity of old pumps and valves so that the water pressure and flow rerouting will be adjusted to the device leakage limitations, in order to prolong the device life-duration. Soil moisture conditions at the surroundings of the valves and pumps, correlated with weather conditions, can assist in understanding if there is a leakage and its severity. Accordingly, the system can control the water pressure and required irrigation flow.

For exposed IoT actuators and sensors, cyber security issues are 100x more complicated, since each of these devices is a vulnerability point into the secured network. Sending misleading information from sensors can trigger a rise in water pressure that will destroy the entire irrigation conduit plumbing.

Using the same sensors can serve different stakeholders. Security aspects will examine historical access, scheduled maintenance and personnel access patterns, as well as overall changes of the system configuration for detecting unauthorized access. Sustainability and maintenance aspects can include comparing soil humidity relative to the irrigation plans, in order to discover irrigation conduit malfunctions, and suggest an infrastructure repair action. Profit system for financial yields can compare the timing and invested resources of water and fertilizers over time, in order to select the less costly operation plan for crops growth. In a contradicting manner, the above security aspect requires rapid CEP response, whereas the financial planning and sustainability one requires slow learning of historical data.

Extension from a single farm unit is done by creating a collective pattern baseline for several connected smart farms, and alerting the grid on potential changes and security risks.

From the end-point IoT pumps and valves that control the real world, through IT security and management aspects, all parts should be considered as key enablers for different types of analytics: descriptive, predictive, and prescriptive. Descriptive analytics will generate reports on the quality of the weather forecast and irrigation planning in correlation with the IT system and device integrity. Descriptive analytics will also indicate the security state, temporarily excluding tampered sensors and actuators in order to employ an accurate control system. Predictive analytics will offer irrigation infrastructure maintenance plans according to availability of data, sensors and actuators, and according to cost of operation for water management savings (i.e., what-if analyses). Prescriptive analytics will automate the irrigation plans according to actual measurements compared to previously analysed systems and will keep evolving according to actual end-to-end component availability.

Descriptive, predictive, and prescriptive analytics vary in the requirements of data sampling rates, control speed for valves activation, enriched data such as GIS location of sensors, and malfunction risk considerations, such as an inactive valve versus a broken water conduit. Consequently, the system situational management should balance and decide on sensor sampling rate, load and content shedding policies, data enrichment, and resulting computation efforts for real time response with optimization and planning considerations. The above motivational example requires both CEP and LK architectures to work in conjunction, and as such, is suitable for a BiDCEP consideration.

6 Discussion and future steps

The BiDCEP conceptual architecture binds both the challenges addressed by CEP and the ones addressed by LK into a unified architectural approach. Specifically, the following challenges are handled:

- Maintaining enterprise IT CEP value.
- Enabling open source-based implementation.
- Controlling data sampling rate and granularity.
- Providing staged process for data enrichment.
- Enabling linear and decoupled scaling.
- Controlling data resolution for consumption.
- Separating data streams based on latency constraints.
- Reducing overall technology footprint and duplicated processing steps.
- Encapsulating and providing data segregation and data cleansing by using separate BiDCEP units in a grid structure.

The concepts presented with BiDCEP are high level ones and assume integration is done with RESTfull API provided by a micro-service container in order to achieve maximum IT flexibility. Not all components must be applied, and overall filter-and-pipe pattern should be applied to the entire streaming process. It is the author's hope that this aggregated conceptual architecture should trigger a technical discussion and practical implementation in different use cases, yielding a construction of an out-of-the-box BiDCEP open-source middleware.

7 References

1. Ait-Sadoune I, Ait-Ameur Y., 2010. Stepwise Design of BPEL Web Services Compositions: An Event_B Refinement Based Approach. InSoftware Engineering Research, Management and Applications 2010 (pp. 51-68). Springer Berlin Heidelberg.
2. Apache Storm http://storm.apache.org/. Retrieved on 12 Dec 2015.
3. Fernandez EB, Ortega-Arjona JL. 2009. The secure pipes and filters pattern. In Database and Expert Systems Application, 2009. DEXA'09. 20th International Workshop on 2009 Aug 31 (pp. 181-185). IEEE.
4. Gal A., Hadar E., 2010. Generic architecture of complex event processing systems. In Principles and Applications of Distributed Event-Based Systems. IGI Global 2010, ISBN

9781605666983, edited by Annika Hinze and Alejandro Buchmann, IGI Global press, 2010. 1-18.

5. Genkin M., 2007. Best practices for service interface design in SOA, Part 1: Exploring the development, interfaces, and operation semantics of services. IBM developerWorks, USA, available at: http://www. ibm. com/developerworks/architecture/library/ar-servdsgn1/,[Accessed 19th March 2010]. 2007.
6. Hadar E. and Perreira M., 2007. Web Services Variation Façade – Domain Specific Reference Architecture for Increasing Integration Usability. IEEE International Conference on Web Services (ICWS 2007), Salt-Lake City, July 2007: 1207-1211.
7. Hirzel M, Andrade H, Gedik B, Jacques-Silva G, Khandekar R, Kumar V, Mendell M, Nasgaard H, Schneider S, Soulé R, Wu KL., 2015. IBM Streams Processing Language: Analyzing big data in motion. IBM Journal of Research and Development. 2013 May;57(3/4):7-1.
8. ITIL, 2015 https://www.gov.uk/service-manual/ operations/service-management.html, retrieved on Dec. 12, 2015.
9. Homer A, Sharp J, Brader L, Narumoto M, Swanson T., 2014. Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications. Microsoft patterns & practices; 2014 Feb 17.
10. Kinley J., 2013. The Lambda architecture: principles for architecting real time big data systems. http://jameskinley.tumblr.com/post/37398560534/the-lambda-architecture-principles-for. Retrieved Dec. 6, 2015.
11. Kreps, J., 2014. Questioning the Lambda architecture. O'reilly Radar. Retrieved 3 December 2015.
12. Martz N., 2011. How to beat the CAP theorem. http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html, Retrieved Dec. 3, 2015.
13. Martz N., Warren J., 2015. Big Data – the principles and best practices of scalable real-time data systems. Manning Publication, 1st Edition, ISBN-10: 1617290343.
14. William A., 2015. Five big technology predictions into 2016. Frost & Sullivan, http://ww2.frost.com/news/ press-releases/five-big-technology-predictions-2016-frost-sullivan/. Retrieved Nov. 23, 2015.
15. Kelly J.E.III., 2015. Computing, cognition and the future of knowing. How humans and machines are forging a new age of understanding. http://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf. Retrieved Nov. 23, 2015. IBM Press.
16. Park Y, King R, Nathan S, Most W, Andrade H., 2012. Evaluation of a high-volume, low-latency market data processing system implemented with IBM middleware. Software: Practice and Experience. 2012 Jan 1;42(1):37-56
17. Precision Agriculture, 2015. https://en.wikipedia.org/wiki/ Precision agriculture. Retrieved Dec. 13, 2015.
18. Reeve A., 2013. Managing Data in Motion: Data Integration Best Practice Techniques and Technologies. Newnes; 2013 Feb 26.
19. Van der Lans R., 2012. Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses. Elsevier; 2012 Jul 25.
20. Williamson L., 2013. Tomorrow's cities: Just how smart is Songdo? BBC News, Seoul, 2 September 2013, http://www.bbc.com/news/technology-23757738