International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Lambda and Kappa Architectures for Data Processing in Healthcare Analytics

Girish Ganachari

Email: *girish.gie[at]gmail.com*

Abstract: ***Healthcare data processing benefits and downsides of Lambda and Kappa architectures. Nate Marz created the lambda architecture to accurately and reliably store massive datasets. Batch, speed, and serving layers do this. Data is stream processed in real time using Jay Kreps' Kappa architecture. The two platforms may enhance hospital operations, decision - making, and patient outcomes. The combination of edge computing, AI, and machine learning optimises systems. This link enables real - time data processing, latency reduction, and security. This study suggests these technologies might change healthcare delivery.***

Keywords: Lambda architecture, Kappa architecture, healthcare data processing, real - time analytics, edge computing, machine learning

1. Introduction

Growing data analytics improves healthcare processes and insights. Real - time analysis enhances patient outcomes and decision - making as healthcare data grows. Analyse wearable devices, genetic, EHR, medical imaging, and socioeconomic database data. Early healthcare systems were tough, therefore real - time analytics are needed. Lambda architecture allows fault tolerance, analysis, service layers, batch processing, and real - time processing, explains Nate Marz. Jay Kreps' Kappa architecture analyses data live. This article discusses how Lambda and Kappa structures might improve healthcare analytics and delivery. A detailed analysis evaluates each design's merits and downsides.

2. Lambda Architecture

Batch layer	1 replicated master node (6 cores CPU, 4 GB memory, RAID-1 storage, 64-bit operating system) 2 worker nodes (12 cores CPU, 4 GB memory, 2 TB storage, 1 GbE NIC) 1 dedicated resource manager (YARN) node (4 GB memory, and 4core)
Speed layer	Shares the Hadoop node
Serving layer	2 nodes (1TB, 4 cores, 16 GB memory)

Figure 1.2: Lambda architecture hardware requirements

Lambda, developed by Nathan Marz, splits difficulties into batch, speed, and serving layers. It handles massive data sets. This architecture covers many data analytics demands, including historical and real - time data. Batch views are dataset summaries created by the batch layer after processing all data. Speed layer assesses real - time input and immediately adjusts system to new information. The serving layer processes historical and real - time data to answer inquiries [1] [2] [11].

a) Applications in Healthcare

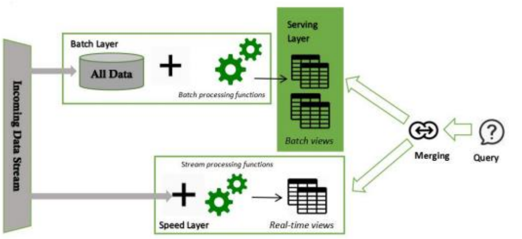

Figure 1.1: Lambda Architecture

Healthcare employs Lambda for real - time and historical analytics. The speed layer analyses real - time wearable device data in patient monitoring to respond quickly to urgent health events, while the batch layer retains EHR data for trend analysis and predictive modelling. A fast heart rate shift may need prompt intervention [11] [22]. Another goal is illness outbreak prediction. ER and social media identify outbreaks in real time. In contrast, the batch layer finds trends in previous data to help public health experts react quickly [3] [12]. The Lambda design is customisable. Genetic, clinical, and real - time health data may help doctors customise therapies. The batch layer finds genetic markers and the speed layer tracks real - time therapy reactions to enhance drugs and patient outcomes [5] [13] [19].

b) Benefits

Batch and real - time data processing is possible with Lambda architecture. Suitable for healthcare applications requiring fast analysis and data integrity [1] [11] [22]. In critical situations, the speed layer handles real - time input, while the batch layer evaluates vast datasets for accurate historical insights [5] [14]. Timely, accurate evaluations enhance patient outcomes. Data is protected by batch layer fault tolerance if the speed layer fails and vice versa [6] [15] [17]. Lambda organises and adapts massive, fast changing clinical and community health data sets using wearables, social media, and EHRs [7] [18] [21].

c) Challenges

Lambda's major limitation is resource - intensive batch and speed codebases. Each layer has its own technologies, concepts, testing, deployment, and development capabilities, making maintenance difficult and costly [1] [3] [12]. Large batch layer data may affect efficiency and consistency. Complex models and algorithms are needed to integrate and synchronise real - time data. Lambda architecture demands plenty of computing power and infrastructure for the batch layer's processing power and storage and the speed layer's low latency and high throughput [7] [15]. Smaller healthcare firms with little IT investment may struggle to deploy these solutions. Both tiers need ongoing deployment and integration, complicating setup and maintenance. Dependability and fault tolerance need debugging, recovery, monitoring, and error handling [7] [18].

3. Kappa Architecture

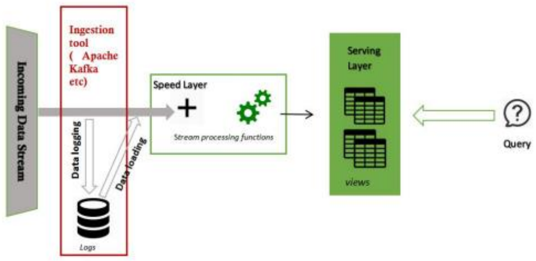

Figure 1.3: Kappa Architecture

Kappa optimises data pipelines without batch processing, says Jay Kreps. Kappa keeps data flowing, whereas Lambda batches and processes. Prioritising real - time processing above data analysis simplifies systems [1] [5] [11]. Kappa's stream processing methodology immediately collects and analyses data for quick insights and solutions.

a) Applications in Healthcare

Kappa architecture handles healthcare data quickly for wearable sensor vital sign monitoring. Critical care, post - operative care, and chronic disease management benefit from real - time anomaly detection [11] [13]. Kappa facilitates telemedicine collaboration, feedback, and decision - making. Doctors may use biometric sensors, video feeds, and medical equipment to treat and evaluate patients remotely [14] [23]. By monitoring heart rates in real time, telemedicine physicians can diagnose and treat patients promptly. Kappa architecture helps doctors detect heart attacks and strokes sooner. Emergency responders deploy more effectively and save lives using real - time data processing [14] [23]. Kappa design in ambulance dispatch and hospital preparation may improve emergency response.

b) Benefits

Ingestion tools	10 servers having each :12 physical processors, 16 GB RAM
Speed layer (Storm)	Minimum one server having : 16 GB RAM, 6 core CPUs of 2 GHz (or more) each, 4 x 2 TB, 1 GB Ethernet
Serving layer	Idem. To Lambda architecture

Figure 1.4: Kappa architecture hardware requirements

The Kappa architectural style places a focus on minimalism in design. System maintenance is easier with stream processing. Eliminating the batch layer simplifies design and reduces development and maintenance costs. Batch and real - time processing no longer need different codebases [1] [14] [23]. This simplified technique speeds up installation, lowers expenses, and simplifies maintenance. Kappa forms facilitate data source adaption and scalability. Since it processes data continually, its design can handle additional data without major changes. This adaptability benefits sophisticated medical facilities and data [16] [24]. Quick growth allows the system to respond to healthcare technology changes. The implementation of simple Kappa software is both faster and less expensive. Low - resource healthcare organisations need minimal infrastructure and deployment. Data processing may enable more healthcare workers treat patients and work effectively [16] [24]. The low cost and financial risk of Kappa architecture benefit testing.

c) Challenges

Kappa architecture has pros and cons. Live data processing may hinder long - term trends and insights. Healthcare practitioners detect trends and explain chronic illness using historical data. Kappa without a batch layer may struggle with historical research or large datasets [1] [7] [11]. Community health and long - term treatment studies may suffer from this constraint. Kappa architecture may violate data integrity and accuracy. Processing several data sources in real time may cause data quality concerns that need extensive error - handling and validation. Due to inaccurate insights and decisions, unreliable real - time data may threaten patient safety [11] [18] [25]. Kappa architecture can handle big historical data and accuracy in real time for clinical and epidemiological studies.

4. Comparative Analysis

Comparing Lambda and Kappa healthcare data processing architectures' pros and cons. Data integrity and historical analysis need Lambda architecture's correctness and fault tolerance. Historical data patterns and predictive modelling benefit from dual - layer batch - real - time processing. These apps benefit from its extensive analytics [1] [2] [11]. Kappa design's simplicity and flexibility enable real - time data processing. This method employs single - layer stream processing for dynamic, time - sensitive healthcare. Data insights are instant [1] [5] [14].

a) Performance Metrics

Most of the time, batch processing makes Lambda architecture superior than Kappa. The batch layer simplifies disease outbreak prediction model data processing and analytics, improving accuracy and dependability [10] [21]. Latency and resource utilisation rise with precision. Resources and batch - speed layer synchronisation may be challenging with two codebases [1] [8] [15]. Kappa design utilises stream processing to conserve resources and accelerate systems. Emergency alarm management and vital sign monitoring are possible with real - time data processing [11] [13]. Its simple processing design may hinder complex analytical tasks that need historical context. Kappa may not be suited for CPU - intensive operations like huge dataset reprocessing and historical data analysis without a batch layer [18] [25].

b) Use Cases

Healthcare applications heavily influence Lambda or Kappa architecture. For data - intensive and predictive analytics, Lambda can handle large historical datasets for long - term trend analysis and treatment effectiveness studies [1] [9] [16]. Durability and great analytics make it ideal for such scenarios. Kappa design offers immediate data insights and responsiveness for telemedicine and real - time health monitoring, making it perfect for alarm and monitoring systems [11] [14]. Staff may respond to real - time data to improve patient care and efficiency. Lambda - Kappa hybrids analyse past and current data. Kappa processing delivers immediate insights, whereas Lambda processing encompasses history [12] [17]. When choosing an architecture, consider accuracy, speed, resource availability, and real - time versus historical analysis [1] [5] [9] [25].

5. Future Directions

Advanced data processing architectures, AI, and machine learning will change healthcare analytics. Lambda and Kappa structure scalability, accuracy, and efficiency should increase with new technology. AI and ML may provide real - time, advanced healthcare insights to address current healthcare concerns [11] [17].

a) Integration With AI/ML

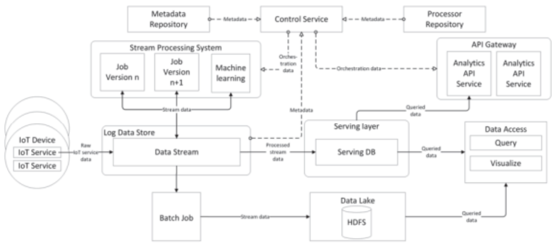

Figure 1.5: Solution proposal and data flows

AI/ML lambda and Kappa designs may improve data processing and prediction. Speed layer Lambda may leverage ML for real - time anomaly detection and predictive analytics. This integration identifies abnormal heartbeats and predicts illness using real - time data [11] [18]. Data aggregation and pattern detection using AI may improve batch layer operations, long - term projections, and trend assessments. Kappa architecture may use AI for real - time data analysis and decision - making.

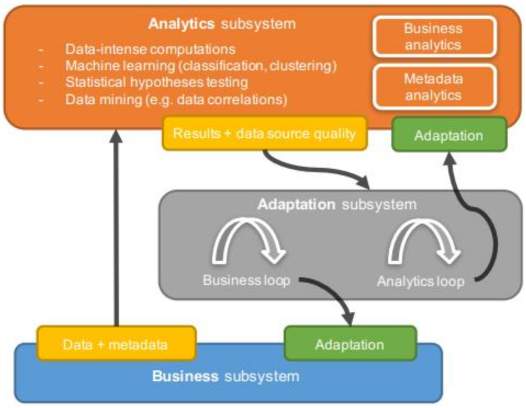

Figure 1.6: Theta architecture

AI systems that adapt to new data inform operations and clinical choices. AI enables telemedicine video consultations, vital sign and patient mood analysis, and quick medical reaction [22] [24]. Rapid treatments and therapy improve with real - time systems.

b) Edge Computing

Edge computing reduces Lambda and Kappa network latency and capacity utilisation by processing data locally [11] [19] [24]. Medical applications like remote patient monitoring benefit from faster data processing and response, improving notification and treatment results [15] [20]. Edge computing allows chronic sickness treatment and real - time wearable alerts without central servers or data latency. Data breaches are reduced by restricting crucial health data network traffic. An AI, machine learning, and edge computing healthcare analytics ecosystem is complete. A quick processing of data, a minimum amount of CPU utilisation, and quick insights [24] [25].

6. Conclusion

Health data management systems have perks and downsides. Lambda's batch and real - time layers make it ideal for reliable data processing in long - term health research and predictive analytics. Kappa stream processing cuts maintenance and increases system flexibility and expandability. Fast data processing and response assist emergency systems, telemedicine, and real - time health monitoring. Design quality will increase with ML, AI, and edge computing. AI and machine learning improve decision - making and forecasting, while edge computing speeds up data processing near the origin. These technologies boost healthcare analytics' versatility and effectiveness. Balance current data processing with past data analysis to choose Lambda or Kappa. Clinical outcomes and healthcare delivery will improve.

References

[1] Sanla, A. and Numnonda, T., 2019, July. A comparative performance of real - time big data analytic architectures. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)* (pp.1 - 5). IEEE.

[2] Zschörnig, T., Wehlitz, R. and Franczyk, B., 2017, April. A personal analytics platform for the internet of things - implementing kappa architecture with microservice - based stream processing. In *International Conference on Enterprise Information Systems* (Vol.2, pp.733 - 738). SCITEPRESS.

[3] Kalipe, G. K. and Behera, R. K., 2019. Big Data Architectures: A detailed and application oriented review. *Int. Journal Innov. Technol. Explor. Eng*, *8*, pp.2182 - 2190.

[4] Zhelev, S. and Rozeva, A., 2017, December. Big data processing in the cloud - Challenges and platforms. In *AIP Conference Proceedings* (Vol.1910, No.1). AIP Publishing.

[5] Zahid, H., Mahmood, T., Morshed, A. and Sellis, T., 2019. Big data analytics in telecommunications: literature review and architecture recommendations. *IEEE/CAA Journal of Automatica Sinica*, *7* (1), pp.18 - 38.

[6] Demertzis, K., Iliadis, L. and Anezakis, V. D., 2019. A machine hearing framework for real - time streaming analytics using Lambda architecture. In *Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24 - 26, 2019, Proceedings 20* (pp.246 - 261). Springer International Publishing.

[7] Brdar, S., Novović, O., Grujić, N., González–Vélez, H., Truică, C. O., Benkner, S., Bajrovic, E. and Papadopoulos, A., 2019. Big data processing, analysis and applications in mobile cellular networks. *High - Performance Modelling and Simulation for Big Data Applications: Selected Results of the COST Action IC1406 cHiPSet*, pp.163 - 185.

[8] Alonso - González, C. J., Pulido, B., Cartón, M. and Bregon, A., 2019. A big data architecture for fault

prognostics of electronic devices: application to power MOSFETs. *Ieee Access*, *7*, pp.102160 - 102173.

[9] Theodorou, V., Gerostathopoulos, I., Amini, S., Scandariato, R., Prehofer, C. and Staron, M., 2017. Theta architecture: Preserving the quality of analytics in data - driven systems. In *New Trends in Databases and Information Systems: ADBIS 2017 Short Papers and Workshops, AMSD, BigNovelTI, DAS, SW4CH, DC, Nicosia, Cyprus, September 24–27, 2017, Proceedings 21* (pp.186 - 198). Springer International Publishing.

[10] Lipčák, P., Macak, M. and Rossi, B., 2019, September. Big data platform for smart grids power consumption anomaly detection. In *2019 federated conference on computer science and information systems (FedCSIS)* (pp.771 - 780). IEEE.

[11] Feick, M., Kleer, N. and Kohn, M., 2018. Fundamentals of real - time data processing architectures lambda and kappa. In *SKILL 2018 - Studierendenkonferenz Informatik* (pp.55 - 66). Gesellschaft für Informatik eV.

[12] Muniswamaiah, M., Agerwala, T. and Tappert, C., 2019. Big data in cloud computing review and opportunities. *arXiv preprint arXiv: 1912.10821*.

[13] Elaggoune, Z., Maamri, R. and Boussebough, I., 2019. The Multi - Agent System Solutions for Big Multi - sensor Data Management. *J. Ubiquitous Syst. Pervasive Networks*, *11* (2), pp.23 - 29.

[14] Palanivel, K., 2019. Modern network analytics architecture stack to enterprise networks. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *7* (4), pp.2634 - 2651.

[15] Zschörnig, T., Wehlitz, R. and Franczyk, B., 2019, May. A Fog - enabled Smart Home Analytics Platform. In *ICEIS (1)* (pp.616 - 622).

[16] Decaneto, A., 2016. Design and testing of an active big data architecture for social and crowding emergency management.

[17] Chatzidimitriou, K., Papamichail, M., Oikonomou, N. C., Lampoudis, D. and Symeonidis, A., 2019, October. Cenote: a big data management and analytics infrastructure for the web of things. In *IEEE/WIC/ACM International Conference on Web Intelligence* (pp.282 - 285).

[18] Preuveneers, D. and Joosen, W., 2017. Access control with delegated authorization policy evaluation for data - driven microservice workflows. *Future Internet*, *9* (4), p.58.

[19] Stolpe, M., 2016. The internet of things: Opportunities and challenges for distributed data analysis. *Acm Sigkdd Explorations Newsletter*, *18* (1), pp.15 - 34.

[20] Demertzis, K., Tziritas, N., Kikiras, P., Sanchez, S. L. and Iliadis, L., 2019. The next generation cognitive security operations center: adaptive analytic lambda architecture for efficient defense against adversarial attacks. *Big Data and Cognitive Computing*, *3* (1), p.6.

[21] Toughiri, R., Wu, X., Ruiz, D., Huang, F., Crissman, J. W., Dickey, M., Froning, K., Conner, E. M., Cujec, T. P. and Demarest, S. J., 2016, October. Comparing domain interactions within antibody Fabs with kappa and lambda light chains. In *MAbs* (Vol.8, No.7, pp.1276 - 1285). Taylor & Francis.

[22] Xu, C., Xu, L., Gao, Z., Zhao, S., Zhang, H., Zhang, Y., Du, X., Zhao, S., Ghista, D., Liu, H. and Li, S., 2018. Direct delineation of myocardial infarction without contrast agents using a joint motion feature learning architecture. *Medical image analysis*, *50*, pp.82 - 94.

[23] Yang, P., Thiagarajan, S. and Lin, J., 2018, May. Robust, scalable, real - time event time series aggregation at Twitter. In *Proceedings of the 2018 International Conference on Management of Data* (pp.595 - 599).

[24] Sarkar, A., Bale, S., Behrens, A. J., Kumar, S., Sharma, S. K., de Val, N., Pallesen, J., Irimia, A., Diwanji, D. C., Stanfield, R. L. and Ward, A. B., 2018. Structure of a cleavage - independent HIV Env recapitulates the glycoprotein architecture of the native cleaved trimer. *Nature communications*, *9* (1), p.1956.

[25] Rimsza, L. M., Day, W. A., McGinn, S., Pedata, A., Natkunam, Y., Warnke, R., Cook, J. R., Marafioti, T. and Grogan, T. M., 2014. Kappa and lambda light chain mRNA in situ hybridization compared to flow cytometry and immunohistochemistry in B cell lymphomas. *Diagnostic pathology*, *9*, pp.1 - 8.