Metamycelium: a domain-driven distributed reference architecture for big data systems

Pouya Ataei[1][0000−0002−0993−3574] and Alan Litchfield[2,3][0000−0002−3876−0940]

[1] Auckland University of Technology, Princeton NJ 08544, USA
[2] pouya.ataei@aut.ac.nz
http://www.springer.com/gp/computer-science/lncs
[3] School of Engineering, Computer and Mathematical Sciences
{abc,lncs}@uni-heidelberg.de

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

The ubiquity of digital devices and proliferation of software applications have augmented users to generate data at an unprecedented rate. In this day and age, almost all aspects of human life is integrated with some sort of software system, that by large, is processing data, and executing the necessary computations.

According to Internetlivestats.com [13] in one second 3,135,050 emails are sent, 1,151 Instagram photos are uploaded, 6,738 Skype calls are made and 147,084 GB of traffic has gone through the internet. That is, in the last minute, 825.04 terabytes have been transferred through the internet.

The rapid expansion and evolution of data from an structured element that is passively stored to something that is used to support proactive decision making for business competitive advantage, have dawned a new era, the era of Big Data (BD). The era of BD began when velocity, variety and volume of data overwhelmed traditional systems used to process that data [1][2].

BD is the practice of crunching large sets of heterogenous data to discover patterns and insights for business competitive advantage [11][6].

Since the inception of the term, ideas have ebbed and flowed along with the rapid advancements of technology, and many strived to harness the power of data. Nevertheless, BD is not a magical wand that can enchant any business processes and many have failed to absorb the complexity of this new field. According to a recent survey by MIT Technology Review insights in partnership with Databricks, only 13% of organizations excel at delivering on their data strategy. Another survey by NewVantage Partners highlighted that only 24% of organizations have successfully adopted BD [9].

Sigma computing report indicated that 1 in 4 business experts have given up on getting insights they needed because the data analysis took too long [4].

Along the lines, McKinsey & Company [?] and Gartner [14] demonstrated that approximately only 20% of organizations have fully adopted BD. These statistics unveil the truth that succesful adoption of BD system is scarce.

Among the challenges of adopting BD, perhaps the most highlighted are "cultural challenges of becoming data-driven", "BD architecture", "data engineering complexities", "rapid technology change", and "lack of sufficiently skilled data engineers" [12][10]. In the past, organizations relied on a few technology giants to account for their analytics and storage needs, whereas today's ecosystem of BD encompasses far-reaching plethora of technologies ranging from visualization to high-level sql-like scripting languages to logging, stream processing and distributed storage.

On the other hand, in react years, more and more companies are shifting to cloud native architectures for improved efficiency, cost reduction, which in turn led to creation of new roles such as chief data officers (CDOs) and chief analytic analytics officer (CAOs) to channel the organizational BD capabilities towards fruition and competitive advantage.

Taking all into consideration, how can one embark on such rather sophisticated journey? what is best initial point for adopting BD ? what can be a good logical approach to address the ever-increasing complexity of BD systems? how can the organization develop a BD system that can effectively handle data of various workloads such as business analytics, real-time stream processing, bulk batch processing and machine learning ?

The initial design, development and deployment of a BD system is only the beginning of BD journey. As system grows larger, data providers and data consumers increase, data variety expands, data velocity extends, and metadata become increasingly more challenging to handle. This means, only a handful of hyper-specialized data engineers would be able to understand the system internal, resulting in silos, burnt out and potential miscommunication.

This creates a perfect ground for immaute architectural decisions that result in hard-to-main and hard-to-scale systems, and raise high-entry blockage. Since ad-hoc BD designs are undesirable and leaves many engineers and architects in the dark, novel architectures that are specifically tailored for BD are required. To contribute to this goal, we explore the notion of reference architectures (RAs) and presented a domain-driven distributed software RA for BD systems.

2 Why Reference Architecture?

To rationalize why we have chosen RAs as the suitable artefact, we first have to clarify two underlying assumptions:

- a sound software architecture is integral to successful development and maintenance of software systems
- there is a substantial body of knowledge in the field of software architecture to support the development of an effective RA

In essence, software architecture is an artefact that aims to satisfy business objectives through a software solution that is evolvable, const-efficient, maintainable, and scalable. In addition, it allows for capturing design issues when they are still cheap. Whereas this practice can be applied to any class of systems, it's particularly highlighted when it comes to design and development of complex system such as BD [1].

Despite the known complexity of BD systems, good news is that a RA can be developed, analyzed and designed to incorporate best practices, techniques and patterns that will support the achievement of BD undertakings. This can help engineers and architect better absorb complexity of BD system development and make it tractable.

This approach to system development is not new to practitioners of complex system. In software product line (SPL) development, RAs are utilized as generic artifacts that are instantiated and configured for a particular domain of systems [5]. In software engineering, IT giants like IBM have referred to RAs as the 'best of best practices' to address complex and unique system design challenges [3]. In other international standardization, RAs have been repeatedly used to standardize an emerging domain, a good example of this is BS ISO/IEC 18384-1 RA [] for service oriented architectures [7]. RAs are used for political speech and even NASA space data systems [8].

Based on the premises discussed above, RAs can be considered effective for addressing the complexity of BD system development for the following reasons:

1. RAs adhere to best practices, patterns, and standards
2. RAs can endow the architecture team with increased openness and interoperability, incorporating architectural patterns that ensue desirable predefined quality attributes
3. RAs can serve as the locus of communication, bringing various stakeholders together
4. RAs can be effective in identification and addressing of cross-cutting concerns
5. RAs can be some of the best approaches to complex system development such as BD, capturing design problems when they are still cheap
6. RAs can take the role of blueprint and summary in the portfolio of software architects and data engineers, resulting in better dissemination of knowledge
7. RAs can manifest the implicit knowledge of software architects as explicit actionable models

3 Research Methodology

There are several approaches to systematic development of reference architectures. In one effort, Cloutier et al demonstrated a high-level model for development of RAs through collection of contemporary architectural patterns and advancements. In another effort, Bayer et al [?]

References

1. Ataei, P., Litchfield, A.: Neomycelia: A software reference architecturefor big data systems. In: 2021 28th Asia-Pacific Software Engineering Conference (APSEC). pp. 452–462. IEEE (2021)
2. Ataei, P., Litchfield, A.T.: Big data reference architectures, a systematic literature review (2020)
3. Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The concept of reference architectures. Systems Engineering **13**(1), 14–27 (2010)
4. Computing, S.: Bridging the gap between data and business teams (2020), https://www.sigmacomputing.com/resources/data-language-barrier/
5. Derras, M., Deruelle, L., Douin, J.M., Levy, N., Losavio, F., Pollet, Y., Reiner, V.: Reference architecture design: A practical approach. In: ICSOFT. pp. 633–640
6. Huberty, M.: Awaiting the second big data revolution: from digital noise to value creation. Journal of Industry, Competition and Trade **15**(1), 35–47 (2015)
7. Iso, I.: Information technology — reference architecture for service oriented architecture (soa ra) — part 1: Terminology and concepts for soa. International Organization for Standardization p. 51 (2016), https://www.iso.org/standard/63104.html
8. NASA: Reference architecture for space data systems (2008)
9. Partners, N.: Big data and ai executive survey 2021 (2021), https://www.supplychain247.com/paper/bi_data_and_ai_executive_survey_2021/pragmadik
10. Rad, B.B., Ataei, P.: The big data ecosystem and its environs. International Journal of Computer Science and Network Security (IJCSNS) **17**(3), 38 (2017)
11. Rada, B.B., Ataeib, P., Khakbizc, Y., Akbarzadehd, N.: The hype of emerging technologies: Big data as a service (2017)
12. Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V.: Critical analysis of big data challenges and analytical methods. Journal of Business Research **70**, 263–286 (2017)
13. Stats, I.L.: Internet live stats (2019), https://www.internetlivestats.com/
14. White, A.: Our top data and analytics predicts for 2019 (2019), https://blogs.gartner.com/andrew_white/2019/01/03/our-top-data-and-analytics-predicts-for-2019/