Cloud Evolution: Enterprise-Wide Digitalization over Clouds

Renyu Yang[†], *Zhao Zhao*[‡], *Jin Ouyang*[◇], *Jie Xu*[†]

[†]*University of Leeds, UK*
[‡]*Imperial College London, UK*
[◇]*Alibaba Group, China*

Abstract

To face the increasingly globalized economy, keep up with the innovation pace and stay competitive, almost every business is aware of the importance to adopt emerging technologies such as cloud, big data, AI to accelerate the digital transformation. However, migrating applications over clouds is a recurring activity and enterprises have to be confronted with unprecedented software maintenance and evolution challenges. This is because strategic and business model heavily determine the methodology of enterprise-wide system design and technical selection considering numerous functional and non-functional requirements. In this chapter, we begin from a technical revisit of cloud evolution and take the Financial Technology (FinTech) application as a representative to conclude the business motivations behind that inherently drive the system design and implementation. To bridge the business requirements and the construction of digital platform, we then present the design paradigm,

referable system architecture and technique selection at different system level. We expect this article can give an instruction to design or rethink the system architecture in the big data and AI era.

1.1 Introduction

Cloud computing and cloud-based services are the buzzwords in the past decade as cloud has been greatly reducing the cost and increasing the operational efficiency. Enterprise developers and managers need to catch the waves of digital disruption and learn how to properly harness it. To keep up with the innovation pace and stay competitive, businesses should be adopting their emerging technologies such as cloud, big data, AI to accelerate the digital transformation. According to Garter's report [1][2], by 2021 over half global enterprises already using cloud today will adopt an all-in cloud strategy.

Typically cloud computing can elastically provide resources in a pay-as-you-go manner and enable enterprises to quickly respond to dynamic customer demands and changing market requirements. There is no need to invest heavily on dedicated software, hardware, and platforms. Cloud technology itself also involutes in the recent past, from the fundamental XaaS business model into a more comprehensive eco-system that includes data-driven collection, storage and processing. With the mature of public cloud service provisioning, it seems more feasible and economical to trust the cloud resources and migrate legacy systems over clouds. For example, cloud computing has been changing the way consumers interact with financial institute such as banks. Cloud computing also increases the turnover of the banks by integrating cost-effective cloud solutions. The banking industry needs to address the ever-growing data input, transactions and payments. Although banks is slowly adopting cloud techniques with concerns such as regulatory and security risks, Cloud technology makes it even faster to scale processing capacity quickly up or down so that they may better react to the changes in customer demand. Through financial data analysis, it is feasible to learn the customers' payment routine, simplify the transaction workflow, reduce the complexity of cash flow thereby increasing the throughput and target the most important customers.

However, migrating applications to cloud-based platforms is a recurring activity and enterprises are confronted with unprecedented software maintenance and evolution challenges particularly in the big data era. In this book chapter, we will firstly revisit the footprints of cloud evolution and review the technique upgrades in the recent past. We then discuss the background business models and motivations that drive the rapid adoption of cloud services on the way of digitalization over cloud infrastructures. Subsequently, we take the FinTech application as an example of large scale enterprise system and discuss the functional and non-functional requirements of building and deploying such a system. Furthermore, we present how to re-design paradigm and propose referable system architecture from very front-end to back-end in order to satisfy the requirements of an enterprise application. In particular, we describe the core components including core business, data management, decision making, data processing, fault tolerance subsystems and how they constitute the internal data-streamline and holistic business workflow. We believe this digitalization procedure is of significant importance and can greatly avoid manifestations of tie-downs by large legacy IT Systems.

1.2 Revisit the Cloud Evolution

Cloud computing and big data have been sparking an unprecedented revolution toward the digital world and information science. In this section we briefly review the progress of different phases.

1.2.1 Early Adopted Model in a Nutshell

The Cloud computing has taken off for a number of years with many new innovations and business applications. When we talked about Cloud computing, we primarily talked about the business model. According to our understanding, the cloud computing is more like a business model than a novel computation model. The Cloud service is maintained by the side of service provider and user just use the software in a pay-as-you-go manner. The usage pattern is just like the way people use any utility such as water and electricity. The underlying infrastructure will be established and maintained by the power supplies and users just pay what they consume.

The most frequently-mentioned key words of Cloud advantages are on-demand elasticity and cost effectiveness. Particularly for the small startup companies, it may be unaffordable to purchase and install all necessary software and hardware in an on-premises way to keep pace with the cutting edge of the market. Cloud computing opens a broad door to those companies and markets.

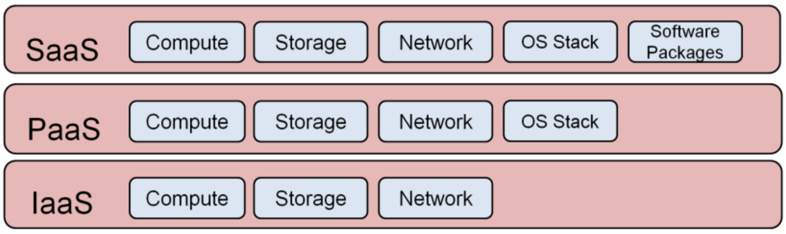

Fig.1 Traditional cloud layers

Basic multi-layered architecture and business model. The cloud vendor**s** offer a variety of service models depending on how customers interact with the cloud resources. From technical perspective, the conventional Cloud are separated into multiple layers (as shown in Figure 1):

- **Infrastructure as a Service (IaaS)**: IaaS provisions the lowest-level building blocks with respect to the computing resource (virtual machines), network resources (software defined network) and storage resources (Cloud storage space). The most distinguished features of IaaS are resource sharing and multi-tenancy. The only thing they need to do is to rent and configure what they want through IP-based connection without dealing with the underlying implementations and complex configurations.

- **Platform as a Service (PaaS):** PaaS providers simplify the developer's effort in deploying and managing applications since they hide all implementation details, such as resource allocation, load balancers, DNS, etc. Intuitively PaaS bridges the development and maintenance of applications and the supplies of all kinds of resources. If we look back from the traditional perspectives of operating systems, PaaS resembles the libraries that can wrap up and release some particular functionality such as process management, memory allocation, database operations, file access. Similarly, the PaaS encapsulates operations of hardware and operating systems in the cloud infrastructure. Representative PaaS services include data storage services such as relational databases, in-memory key-value storage, distributed file systems, and messaging system that can publish and subscribe messages, etc.
- **Software as a Service (SaaS).** SaaS is an approach of delivering software applications in a service mode based on user's subscriptions. It is sometimes the most direct access way by users. End-users do not need to install and prepare any execution environment on their personal computers. Typically users can access the SaaS by using a thin client via a web browser to handle the task at hand. Examples of SaaS applications can be found everywhere, like the web-based email, Office 365, Google doc, etc. Typically, you can use SaaS application as long as the Internet is accessible.

1.2.2 Development: Driven by Big Data

With the mature of computation as a service that can fundamentally meet the basic infrastructure capacity, the architecture evolution is primarily driven by the big data business and the increasing demands of big data processing. Consequently, the emerging big data ecosystem became the other side of Cloud computing.

Figure 2 illustrates the multi-layered architecture that best depicts the bird-eye picture of big data applications and researches. In the data storage layer, data are stored in different entities according to the structure and use scenario. In general, system logs or transaction logs will be archived into the buckets of distributed file system. Batch processing such as word count and field ranking are the representative applications based on the distributed file systems. SQL storage is the most common and long time applications in the data management area. Especially for the traditional information system, SQL like store and queries based on this are the most important step in digitalizing business workflows and data management. In this case, the rational database management systems (RDBMS) such as MySQL, SQLServer, Oracle, etc. are the representatives.

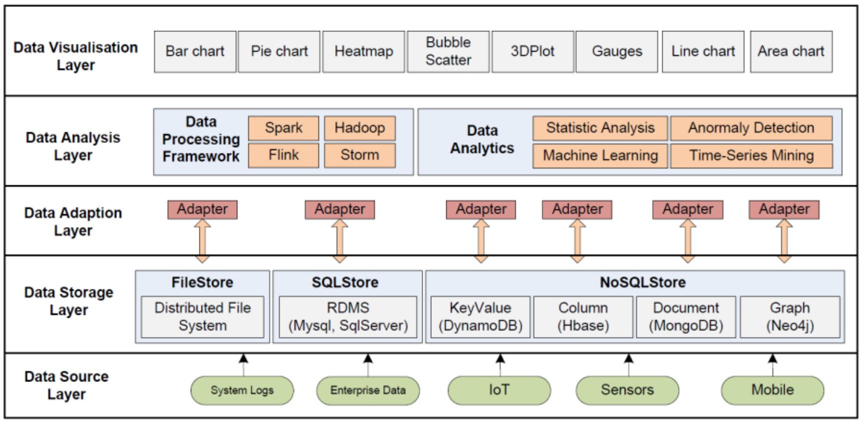

Fig.2 Big data multi-layered architecture overview

With the advanced data science, data diversity and storage types begin to increase. The concept of NoSQL [18] is raised at this moment. The data in RDBMS will be stored and indexed according to rows. However, column-based or object-based approaches are applied in NoSQL store. Another distinguished difference between them is RDBMS systems are aimed at relational data that need complex querying, whereas NoSQL systems is aimed at processing an enormous amount of mostly non-relational data. For instance, objects such as key value pairs or nested key value pairs should be empowered by optimized storage scheme rather than purely relying on the RDBMS. The structure of different NoSQL databases may differ - e.g., columns in HBase [3] or JSON in MongoDB [4]. Entire documents in the form of JSON objects can be regarded as nested key-value pairs. Additionally, JSON also supports arrays and understands different data types, such as strings, numbers and boolean values. However, simple key-value stores don't support nested key-value pairs. In comparison, HBase store the data according to columns, thus make it suitable for operations based on data pertaining to the same column. Archived data analysis such as tracelog is the most typical scenario. Accordingly, there are many cloud products and services that provisioning public data as a service (DaaS). According to the bespoken concept, DaaS is a sub-category of IaaS which particularly provisions the multi-tenants data relevant services. Figure 3 demonstrates representative data-relevant products in giant Internet companies such as Google, Amazon, and Alibaba.

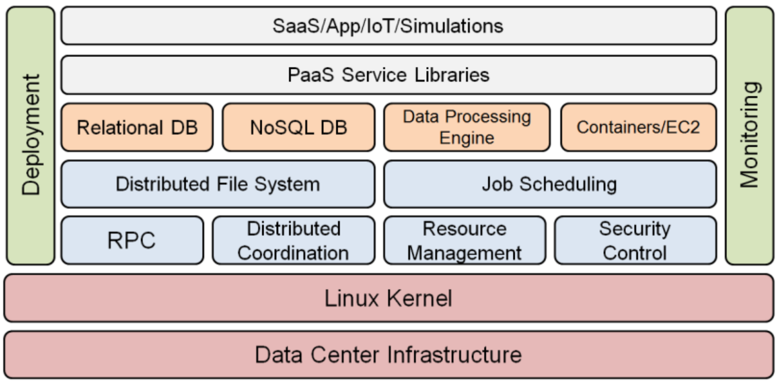

Fig. 3 Data-centric product layout.

1.2.3 At the Minute: Large-scale Commercialized Clouds

Particularly with the rapid development of Cloud computing, when we revisit the overall revolution over the past decade, we believe that more businesses will leverage cloud to implement their applications and make their market strategies.

Looking into the second decade, Cloud computing is becoming mature on several fronts and objections to Cloud technologies are significantly lessening. Some business and technical innovations are only made in the format of Cloud services. In this context, the quality of service aspects such as system reliability, platform-level security, operational costs are becoming dominating considerations as the pressure to move to Cloud service rapidly soars. Another urgent requirement is to bridge the Cloud infrastructure with big data applications. One of the most distinguished advances of Cloud computing is the data de-duplication and thus facilitates easy use of data. Cloud storage can reduce the data transfers and optimize the data placement among geo-distributed locations. Specifically, such transfers are increasingly infeasible since the size of datasets have already grown TB or PB level. However, in the Cloud, as long as the service is accessible through Internet, the same data can be accessed and manipulated remotely. The multi-replica scheme designed by Cloud storage can guarantee the reliable data access with consistency, thereby liberating users from manual storage and synchronization.

Despite the advancement of Cloud techniques and increasing pressure to move applications to Cloud services, not all incorporations and their projects were willing to leverage Cloud services due to security issues or regulatory concerns. However, those concerns are not necessary at the minute. In fact, with the mature of Cloud techniques, Cloud services have been increasingly regarded more secure than on-premises applications according to recent reports [2]. Commercialized clouds and hybrid cloud markets are making the data utmost secure and the system with high availability. Moreover, the legal and regulatory compliance in different industries will increasingly ensure where cloud resources are located, where data is stored and who have the rights to access it. Therefore, it is no wonder to expect many industrials and enterprises to embrace Cloud computing in order to achieve increased agility, reduced costs, and gained value. For instance, many of the most popular and successful Internet services including Netflix, Instagram, etc. are heavily making use of commercial clouds.

Nevertheless, migrating applications to more cloud-based platforms is a recurring activity and enterprises are confronted with unprecedented software maintenance and evolution challenges particularly in the big data era. The fundamental concepts of specific business models did influence the design and implementation of cloud systems, so we indeed to carefully identify and summarize what are the basic requirements by diving into business models.

1.3 Emerging Internet Applications: Business Models and Requirements

In this section, we firstly analyze the most important motivations for enterprise to digitalize their work process and information systems from business perspectives. Banking and finance industry, one of the most traditional and vital service in citizen's daily life, are experiencing unprecedented digital transformation due to the radical changes of financial market and cash flow. In this context, substantial Internet applications such as *FinTech* [20], *Blockchain-based* [21] *financial supporting systems*, etc. are rapidly emerging in the finance realm which completely change individuals' lifestyle. To follow up, we intend to enlighten readers as to what the requirements are and how to leverage proper cloud techniques to achieve so.

1.3.1 Business Motivations for Digitalization over Clouds

It is no secret that migrating to clouds significantly helps take advantage of the boosting economies and optimizes costs by reducing the total cost of ownership. In particular, the main motivations of enterprise-wide digitalization are:

Minimized cost. Reducing the margin is always the top topic of business for getting continuous funding and investment. Any innovations or process improvements serve the ultimate goal. Compared with building a dedicated computing pool (e.g., data center cluster), it is apparently far more economical to run applications on the clouds. Cloud services can also be adjusted to match the seasonal changes in business, indicating the great amount of usage elasticity.

IT legacy estate governance. A number of industries such as banking, insurance, and transportation still rely on outdated implementation techniques, stale databases but still serve a great number of customers 24 hours. This occasion leads to the fact that such systems are extremely difficult to be replaced due to they are required to deal with vital business processes. The risk of allowing IT legacies for existing will aggravated with the increments of hardware and software aging [9][10]. Thus, it is high time to pay down technical debt and modernize the underlying implementations.

Improved agility and rapid revolution. Enterprises are facing up to market changes and customer surges. It could be catastrophic if their information system or front-end services were not able to tackle variable demands. Therefore, quick and even proactive response to potential scalability and resizing issues is significantly important. Due to the elasticity and on-demand resource providing of cloud computing, cloud is naturally the strategic enabler to deal with auto-scaling problems.

1.3.2 Example - FinTech Applications

The digital evolution has been sweeping the financial industry such as business to business, online payment, digital banking and so forth. This is inevitable due to the ubiquitous computing power, massive storage, network connections and powerful data analysis empowered by the Cloud and big data.

FinTech (Financial Technology) is referred to the innovative use of computing technology in the design and delivery of financial or banking services. Many people have already regarded FinTech as the most important example of technical disruption. The cash chain and capital supplies can be easily turned-over through FinTech. The particular format of FinTech may include: digital banks, P2P lending platforms, personal finance management, payment and billing software and mobile applications, etc.

However, FinTech solution needs to be integrated with existing IT systems rather than replacing the old ones. This will raise a storm which involves legal, financial expert and computing architects and developers to think about a platform-level journey to the cloud by means of either cloud-based redesign or cloud-oriented migration. It can leverage cloud-based architecture and data-driven approach to improve the daily financial functionalities such as payment, lending, borrowing, and investing, etc.

1.3.3 Functional and Non-functional requirements

Function Requirements. Take the FinTech application as an example again. The technology enable people to transfer moneys with only one-click, manage their portfolio from their own home, and make online transactions by their smartphones. To implement these functionalities, there are specific requirements ranging from front-end application design to the back-end data storage and data-driven analysis infrastructure.

- **Front end.** To deal with the great number of user submissions such as report fetching, balance query, etc., efficient data caching for client end and mobile application is very important. In-memory cache is typically used to store user specific session for the sake of reduced latency.
- **Middle end.** The core requirements are data routing and load balance. There is a challenge of scalability when the number of users and transactions is constantly growing and the middle-end system needs to tolerate the high loads. Moreover, to connect the front end and the back end and support the data display, there are many relational data structures that need to be stored and queried. Thus concurrent database access needs to be guaranteed at the middle-end.
- **Back-end.** The primary task at the back end is to store core transaction data for pattern mining, archived records for global compliance assessment and a large amount of system tracelog for the business optimization. The characteristics of these data are: super large-volume, non-structured and immutable. It is noteworthy that for the purpose of

global compliance and tracking, the secure information and billing should be protected from any attacks and deliberate modifications.

Non-functional requirements. Apart from the functional requirements, there are a large number of non-functional requirements:

- **Effective and reliable transaction.** FinTech businesses have to primarily deal with contradictory challenges - the transactions have to be quick but reliable. That is, we usually need to strike the balance between the end-to-end transaction time and the accuracy. High loads even make the task more complicated.
- **System security.** Cryptocurrency hacks, DDoS attacks and other vulnerabilities, like cross-site scripting (XSS) [19] and SQL injection attacks are the most common threats to FinTech applications. A large sum of money could be easily stolen due to a security bug. Maintaining the confidentiality and security of financial information of customers and company data is very important.
- **Multi-tenancy security.** Multi-tenancy is one of the Cloud characteristics and several customers share a virtual instance of a software application or a unit of system resources. However, a security challenge associated with it is how to prevent one customer from accessing another customer’s data and resource states. Data isolation and interference are the main sources of risks. How to mitigate such risks at all kinds of level such as access control, file system isolation, resource split and boundary control is an open challenge.
- **Runtime Reliability.** It is highly desirable to timely detect the possibilities of abnormal signaling and system failure. Particularly for the FinTech systems, you need to actively monitor your systems and alert when some abnormal cash flow or transactions manifest.

1.4 Migration towards Clouds: How Business marries Technology

This subsection mainly discusses how the business marries the technical solutions. Firstly we give an overall system design of digital FinTech platform based on business requirements, followed by the corresponding technical architecture and the detailed solution.

1.4.1 FinTech System Design - From Business Perspective

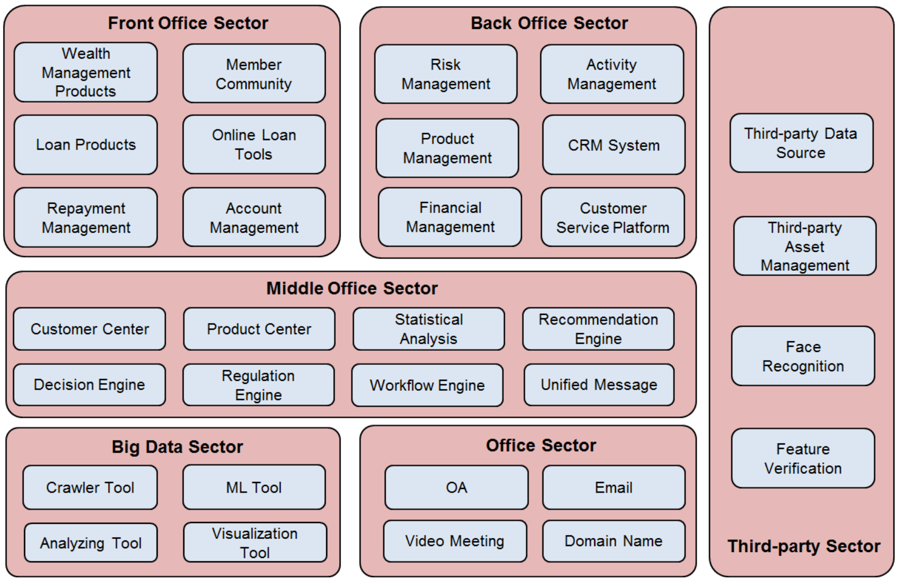

Fig.4. FinTech System Overview.

Figure 4 illustrates the overall design division from business perspective. **Front office Sector** offers users online banking application that allows them to purchase different financial products such as wealth management or loan products, manage their accounts, view historical transaction records, pay bills, etc.

Back office Sector facilitates daily business by providing basic management tools such as risk management, product management, financial management system, customer relationship management system (CRM system), etc. For each deal, CRM system helps keep fundamental user information up to date, track business interactions, synchronize data across systems to maintain and improve quality relationships with customers. While deal information flows into risk management module, product managers can then evaluate the involved potential risks by deep diving into user account information and credit ratings. This assessment provides real-time account position values and risk exposure, thereby showing flags near suspicious transactions and alerting administrators. Every deal will be eventually reflected in financial management system which creates, links, stores and reports financial information of different transactions. Other tools such as product management provide product design, review and approval; product release and activity management need to show every system events and history of account activities.

Middle office Sector bridges the gap between front office and back office. It covers customer center, product center, statistical analysis, recommendation engine, decision engine, etc. Customer center, for instance, communicates with customers, answers their queries, solves and records their complaints.

Big data Sector covers a wide range of tools including crawler, analyzing, machine learning and visualization. By automating the data analysis and trend forecasting, machine learning algorithms can offer solutions to better detect and prevent fraud, target the customer sophisticated demands, interpret documents, etc., which creates a new level of financial services. Visualization tool, on the other hand, presents data in a more interactive and engaging manner during communications.

Office Sector includes OA, video meeting, email and domain name to help information transfer, user interaction, document, data storage, and so on.

Third-party Sector employs information from third-party, covering third-party data source, third-party asset management, face recognition, etc. Data from third-party could help Fintech companies to portray users' credit status and conduct a more thorough and accurate risk analysis, and better analyze user behavior. Third-party management mitigates the regulatory arbitrage and cross-border arbitrage derived from cross-border business model of Fintech.

1.4.2 Architecture and Implementation

The business of banking and finance will generate thousands of millions transactions at any time and it is quite critical for FinTech to leverage this data in order to establish a competitive advances within the market thereby making instant and intelligent business decisions.

Figure 5 demonstrates the referable system architecture for a cloud-based FinTech platform. Through security enhancement appliances such as firewall and DDoS (Distributed Denial of Service) protector, the system can be ultimately protected from security threats [16]. Afterwards, the data stream and network flows can safely arrive at the end of internal system components. For scalability consideration, several load balancers are set up in front of the virtual computing resources that are used for underpinning specific business systems. To guarantee the user experience, CDN [17] appliances are deployed onto geographically close locations to form a high-speed proxy service. To support diverse requirements of data access, a variety of databases appliances are placed among different subnets. The databases will be connected with the business modules and internal audit and assessment system, providing data pathways to achieve data-based audit and compliance control. Specifically, there are five primary subsystems that constitute the overall architecture:

Firewall. The goal of the Information security is *confidentiality*, *integrity* and *availability*. Not everyone should be allowed to login the server and service through ssh. Particularly, firewall can block traffic intended for particular IP addresses or server ports. Only individuals who have proper credentials can be allowed the access via a secure connection, typically a virtual private network (VPN).

Load Balancer. LB is not only to manage the traffic flows across different enterprise regions, but to enhance the system security as safe guard and

DDoS detector. DDoS attacks are the most common threats to both public cloud providers and any size of service suppliers. The load balancers are typically displayed in front of the middle-end and back-end system components to reduce the dramatic traffic flows. The LB should also be able to auto-scale its instance number in accordance with the change of both user submission rate (seasonal or spiking traffic) and the back-end service (downtime or recovery events).

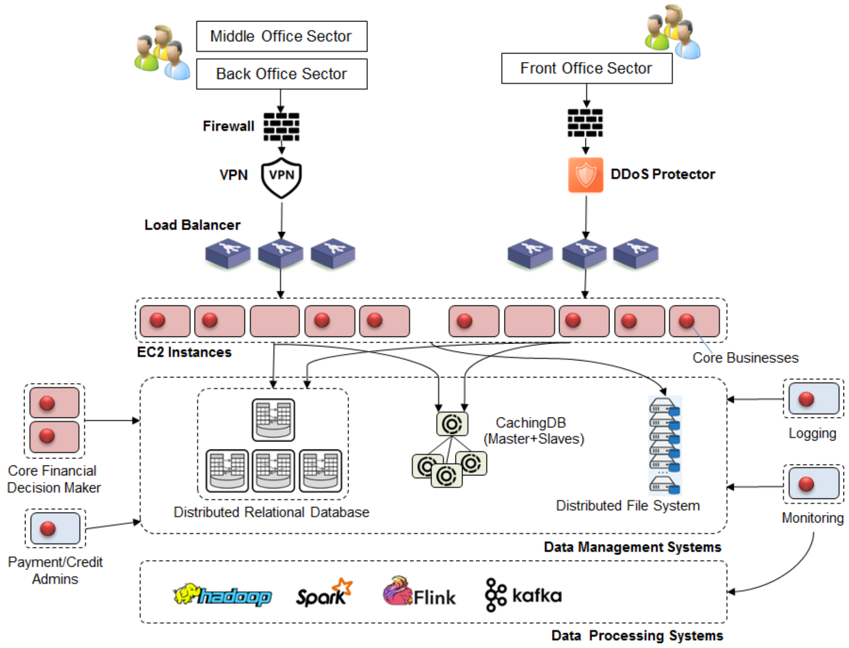

Fig.5 System architecture for a Fintech application.

Core business sub-systems. The modules in terms of business functionalities such as account management, lending supplies and demands matching, mortgage product information system, investment product, etc. The basic requirements for such subsystems should be CRUD operations and transaction operations with very high consistency guaranteed. Those business relevant systems are deployed and maintained in computing servers due to those appliances are resource-intensive. The ideal way to do this is to create multiple-replica virtual machines or Docker containers within dedicated infrastructures in enterprise domain or rent public virtual machines over public clouds. For instance, AWS's EC2, Google's Compute Engine, Microsoft's Azure, Alibaba's ECS or containers provided by open-source Kubernetes are realistic and optional candidates.

Data management sub-systems. The data access is the essence and highly likely to be the performance bottleneck in the information systems. Rapid and reliable data access and query is the basis of high-quality user experience. This is particularly true for FinTech clients whose operations are all relevant of money and properties. The main objective for the data management is to support the data networking with capable data pipeline (both up-streaming and down-streaming) and to provision concurrent data queries. To support different types of data format and use case, both relational (SQL) and NoSQL database technologies matter and should be investigated without any ignorance.

Traditional relational database is still important due to a large amount of structured data need to be stored and queried every day. The structured data widely manifest in the banking area such as name, billing address, occupation, purchasing records, credit records etc. More importantly, the join queries among different data tables play a very vital role in exploiting the correlative behavior and bridging the gaps between different entities in the virtual financial world. SQL database query also outdoes other solutions in terms of interactive response time and integrity of data which makes it much easier to meet compliance regulations.

This does not mean NoSQL is pointless. Reversely, with non-structure data is growing fast, NoSQL solutions are demanded to face documents, graphs or key-value data. While no single database provides multiple comprehensive capabilities of integrating and accessing data, the hybrid solution should be the answer of effective data management. No matter which technologies are chosen, they should meet the highest level of authorization and full ACID transactions. In-memory key-value store automatically replicates and partitions data caches across multiple nodes and deliver elastic scaling to add or remove nodes.

Another challenge that needs to be tackled is the scalability. To support high concurrent data access, the distributed architecture with master and multiple slaves is the widely-accepted design paradigm. For instance, to ensure the ACID, a slave can act as the read-only agent whilst only the master is permitted to write data into the database. There are abundant production-level cases such as Redis, Mongodb, AWS's DynamoDB, Google's Cloud Bigtable, Alibaba's OSS and OTS.

Core financial decision sub-system. There are many decision procedures such as credit limit management, risk identification and collateral estimation, etc. Auto-biding is a good example to illustrate what is the decision making system in FinTech. P2P funding and lending is the most typical scenario that auto-biding serves. Auto-bid can automatically match the demands from consumer lenders with the investors who can offer their money. Theoretically, it is the perfect matching or maximum matching problem for bipartite graph, but this procedure still needs to take into account the transaction risk and other regulations. For instance, the FinTech platform should spread risk for its investor whereas it needs to avoid a lender over-locked by one borrower during the loan selection. Therefore the decision making is NP-hard and time-consuming. A separate computing module is usually established to achieve the secure and effective calculation. Similarly, it would be better to deploy them in a secure and isolated environment such as security-enhanced virtual machine or containers.

Internal Administration subsystems. The main means to detect potential risks and audit the system behaviors are tracing records and data analysis. Additionally, this procedure should be secure and internal without any external interference from customer and operators. This can be achieved by VPN tunnel with encryption. Therefore, paving the way from data provisioning sector to the administration sector and streamlining the data processes are very essential. The query engine that can automatically adapt and configure different data sources should be able to offer such capabilities. The success also reckons on the big data processing to exploit deeper observations in terms of transactions, account behavior, balance cash flow, etc.

Batch processing and stream processing sub-systems. Batch processing is the procedure of processing data that have been accumulated over a period of time whilst the streaming processing needs to return the analysis result in real time. In FinTech, a large number of transactions need to be revisited. The process is typically performed at the end of a day or week in order to get a statistic pattern or user behavior. Additionally, to guarantee real-time security and detect frauds, stream processing can be used to detect anomalies when some signal or metrics manifest. Available stream processing platforms such as Apache Kafka, Apache Flink [5] and Apache Spark [6] have already become a must for big data family. Spark framework can provide APIs to perform massive distributed processing over resilient datasets. This is especially the case for small FinTech companies who may not have a dedicated data infrastructure in place to build the whole data-relevant eco-system.

Monitoring sub-system. There is no secret the importance of monitoring system is. This is the fundament for the anomaly detection and full stack alerting. To implement the monitoring system, we firstly need to make sure some performance counter or key function metrics at different system levels can be added and exposed to a message system. For instance, Telegraf [7] is a daemon that can run on any server and collect a wide variety of metrics from the system (cpu, memory, swap, etc.), common services (mysql, redis, postgres, etc.). Kafka[9] is a promising message framework to solve growing Fin-Tech data problems - collect and enrich data to generate more insights. How to timely store the collected data is the next question we have to answer. Time-series databases are of particular interest and influxDB [8] is a popular open source distributed database. The above software stack allows us to easily integrate InfluxDB into their existing monitoring and logging stack, and to continue using their existing tools that consume Kafka data.

Access control and multiple tenant isolation. Because tenants share the same underlying systems and hardwares, a run-away process or an overload situation incurred by a tenant may have negative impact on other tenants. Robust system monitoring can quickly detect and mitigate the potential degradation. A separate module of access controller is typically meaningful to perform per-tenant rate-limiting throttling and resource quota management.

1.4.3 Fault Tolerance Considerations

It is essential to provide high availability features such as platform-level replication, automatic failover, fault tolerance, and rapid recovery at an enterprise-wide scale. The most common way to reach fault tolerance is redundancy. Fortunately, Availability Zones (AZs) are widely designed and adopted by main cloud providers. AZs are unique fault-isolated physical locations with independent power, cooling and network infrastructures. Each AZ consists of one or more datacenters and houses infrastructure to support mission critical applications with fault tolerance [12].

Deploying a subsystem as multiple-replications across different regions or management domains can no doubt handle the irresponsive service or slow processing when some of the replications are under heavy load or experiencing transient software or hardware failures [13]. Across multiple zones design is a demonstrative case. For instance, system developers should create zone-redundant load balancer and SQL database while planning subnets for both front-end subsystems and the back-end databases. For critical business modules, wherever they are deployed, the virtual machines or containers should be spread over at least three available zones. In addition to component-level fault tolerance, a dependable network increases business continuity. The network module that underpins the whole business should design the network connections with diverse pathways, and regional layers of redundancy.

1.5 Next Decade - Cloud Evolution

Financial institutions are increasingly keen on the automatic investments and trade which reckon more on algorithmic and data-driven conclusions rather than direct human decisions. Therefore, technical staff should be aware of data Science, AI, Blockchain, etc. This raises more challenges to both cloud providers and cloud consumers: it is important for enterprises such as fintech institutes that are re-architecting with next generation operating systems to understand the cloud if they want to extend the efficiency of legacy architectures and capture the immense digital economic market.

Cloud will redefine all digital solutions at new benchmark level and it will be much more intelligent and connected than ever before. However, there is no wonder that cloud will no longer be regarded as the buzz technology because this technology has become commonplace. Terms such as cloud and digital will not really matter due to their pervasiveness in the IT world. The focus has been transited onto the customization and intelligence.

Flexible, customized and smarter clouds assisted by Artificial Intelligence. Although we discuss in this article how to seamlessly redesign and migrate legacy enterprise systems into the cloud-based infrastructures, several factors like external regulation and compliance requirements still imped some applications from migrating to Clouds. However, the big advances of cloud capability should not be ignored. Enterprises such as FinTech companies will be creating great opportunities for data analysis and business insights. Literally, Data Science and Artificial Intelligence can help solve the core problems in risk assessment, identifying customer transaction habits, fraud prevention, etc.

Collaborative Clouds. If a banking service is international and has operations in different time zones, data storing and processing is fairly time-consuming and conflict-resolving with a great number of synchronizations. Just like the airline alliance, Joint cloud model [15] and Fog orchestration model [14] might be useful: Local cloud vendors will delegate and serve the local customers and they collaboratively work together to deal with payment, billing and other financial services.

1.6 Conclusions

In this chapter, we take the Financial Technology (FinTech) application as a representative technical disruption and explore the underlying business motivations that inherently drive the system migration and evolution over Clouds. We rethink the system architecture and design paradigm in order to satisfy business requirements and the transformation of digital platform. The main conclusions from our research can be summarized as follows:

More businesses will leverage cloud techniques to implement and deploy their applications. Applications with SaaS mode will continue emerging as the mainstreams. However, new concerns such as compliance, vendor lock-in and data sharing will be on the fly.

The success of cloud evolution and application migration should best fit with the capabilities and target the customer requirements. Choosing who to go after, which system modules to improve and which techniques to adopt will be highly important.

Containerisation will become the next trend to maximize the profits of devOps. Container, as the light-weighted virtualization technique, will be more widely adopted in resource management, service orchestration and multi-tenant isolation and control.

Acknowledgement

This work is supported by the National Key Research and Development Program (2016YFB1000103) and the National Natural Science Foundation of China (61421003). This work is also partially supported by Beijing Big Data and Brain Computing (BDBC) innovation center. Zhao Zhao is the corresponding author of this book chapter.

Reference

1. Four Trends in Cloud Computing CIOs Should Prepare For In 2019. https://www.forbes.com/sites/forbestechcouncil/2018/07/05/four-trends-in-cloud-computing-cios-should-prepare-for-in-2019/#18b1c6684dc2
2. Cloud Computing enters its Second Decade. https://www.gartner.com/smarterwithgartner/cloud-computing-enters-its-second-decade/
3. Apache HBase. https://hbase.apache.org/
4. MongoDB. https://www.mongodb.com/
5. Apache Flink. https://flink.apache.org/
6. Apache Spark. https://spark.apache.org/

7. Telegraf. https://www.influxdata.com/time-series-platform/telegraf/
8. InfluxDB. https://www.influxdata.com/
9. Huang, Yennun, Chandra Kintala, Nick Kolettis, and N. Dudley Fulton. "Software rejuvenation: Analysis, module and applications." In ftcs, p. 0381. IEEE, 1995.
10. Grottke, Michael, Rivalino Matias, and Kishor S. Trivedi. "The fundamentals of software aging." In Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on, pp. 1-6. Ieee, 2008.
11. Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing." In Proceedings of the NetDB, pp. 1-7. 2011.
12. Chaczko, Zenon, Venkatesh Mahadevan, Shahrzad Aslanzadeh, and Christopher Mcdermid. "Availability and load balancing in cloud computing." In International Conference on Computer and Software Modeling, Singapore, vol. 14. 2011.
13. Garraghan, Peter, Renyu Yang, Zhenyu Wen, Alexander Romanovsky, Jie Xu, Rajkumar Buyya, and Rajiv Ranjan. "Emergent Failures: Rethinking Cloud Reliability at Scale." IEEE Cloud Computing 5, no. 5 (2018): 12-21.
14. Yang, Renyu, Zhenyu Wen, David McKee, Tao Lin, Jie Xu, and Peter Garraghan. "Fog Orchestration and Simulation for IoT Services." (2018).
15. Wang, Huaimin, Peichang Shi, and Yiming Zhang. "Jointcloud: A cross-cloud cooperation architecture for integrated internet service customization." In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pp. 1846-1855. IEEE, 2017.
16. Sabahi, Farzad. "Cloud computing security threats and responses." In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, pp. 245-249. IEEE, 2011.
17. Pallis, George, and Athena Vakali. "Insight and perspectives for content delivery networks." Communications of the ACM 49, no. 1 (2006): 101-106.
18. Han, Jing, E. Haihong, Guan Le, and Jian Du. "Survey on NoSQL database." In Pervasive computing and applications (ICPCA), 2011 6th international conference on, pp. 363-366. IEEE, 2011.
19. Vogt, Philipp, Florian Nentwich, Nenad Jovanovic, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. "Cross Site Scripting Prevention with Dynamic Data Tainting and Static Analysis." In NDSS, vol. 2007, p. 12. 2007.
20. Philippon, Thomas. The fintech opportunity. No. w22476. National Bureau of Economic Research, 2016.
21. Underwood, Sarah. "Blockchain beyond bitcoin." Communications of the ACM 59, no. 11 (2016): 15-17.