Script Based Migration Toolkit for Cloud Computing Architecture in Building Scalable Investment Platforms

Rao Casturi[1(✉)] and Rajshekhar Sunderraman[2(✉)]

[1] V.P. Risk Management, Voya Investment Management, Atlanta, GA 30327, USA
Rao.casturi@voya.com
[2] Department of Computer Science, Georgia State University, Atlanta 30062, USA
raj@cs.gsu.edu

Abstract. The 2008 Financial Crisis which created a global financial market meltdown is mainly due to badly structured mortgage loans with poor or subpar credit quality and lack of proper tools to measure portfolio risks by the lenders. Even though several problems led to this crisis, we looked at this from a Big Data. Had the infrastructure and analytical analysis tools were present to the lenders, they would have found the various early warning signs on these mortgage loans and could have better prepared for the crisis. Aftermath of the crisis, all the big financial institutions took a fresh look and embarked onto build various tools and frameworks to address this Big Data in their portfolios with data driven analysis. The 3Vs (Velocity, Volume and Variety) of the Big Data in our Mortgage Loan Analysis System challenges our traditional approach in collecting, processing and presenting the individual and aggregated loan level data in a meaningful format to facilitate our portfolio managers in decision making. The traditional methods are implemented on a standalone on-premises SQL server. Our Framework creates the foundation of migrating from traditional standalone database architecture (on-premises) to Cloud Computing environment using "Script Based Implementation". The methods we present are simple but effective and saves resources in terms of Hardware, Software and on-going maintenance costs. Big Data "Capture, Transform, Calculate and Visualize" (CTCV) implementation takes a phased approach rather than a big bang model. Our implementation helps the Big Data Management to be part of organizational tool kit. This saves hard dollars and brings us in line with the overall firm strategic vision of moving to Cloud Computing for Investment Management Services.

Keywords: Big Data · Financial applications · Cloud Computing

1 Introduction

In any investment portfolio management, diversification of the portfolio holdings is critical to achieve client expected returns by minimizing the downside risk in the portfolio. Diversification can be within a specific sector or across sectors or different

M. Elloumi et al. (Eds.): DEXA 2018 Workshops, CCIS 903, pp. 46–64, 2018.
https://doi.org/10.1007/978-3-319-99133-7_4

types of fixed income bonds. One such investment strategy is to have exposure to the mortgage bonds in the portfolio along with other investment instruments [1]. The total Mortgage Debt Outstanding for 2017Q3 is estimated to be around $14.7 trillion in U.S. The structure of these mortgage securities are called pools and they consists of mortgage loans taken by individuals. The raw data (factor data) is released on a monthly basis by various Mortgage Agencies like Ginnie Mae, Freddie Mac and Fannie Mae. The individual financial institutions either use third party vendor provided solutions or build their own data gathering applications to collate this information. The data set can be around 2 TB per month and grows from month over month as the loans and pools tend to increase as time progresses. For smaller institutions the cost in implementing third party vendor for Big Data solutions are expensive and can't justify the costs. The other problem is presenting this information in a useful and flexible manner for the portfolio managers. We were able to solve our Big Data problem by breaking it into smaller manageable phases and build a modular based frame work to save organizational costs and reduce maintenance and other infrastructure overheads. Our approach (CTCV Framework) gave our portfolio managers the ability to analysis huge amount of data in a very short period compared to the legacy EXCEL based application. The EXCEL based model was severely limited to the amount of information they can analyze in a given day due to the technical limitations of processing Big Data in EXCEL.

The current paper is organized into eight sections. Section 2 introduces the preliminary definitions and background information. Section 3 highlights the problem, presents related work done on migration of traditional database models to cloud based models in academia, and introduces our proposed solution. In Sect. 4, we present the solution and in Sect. 5 shows the implementation. Section 6 captures the results and Sect. 7 is conclusion with Sect. 8 laying the foundation for our future work.

2 Preliminaries and Background

The Investment Portfolio consists of Fixed Income Bonds [1] and or Equity securities. In the day and age of analytics, the key analytical indicators [2] of these investment assets drive lot of portfolio decisions. Calculating and picking up trends in these analytical measures is a challenge when we need to go through several millions of mortgage loans. Further the Bonds can be classified into corporate and mortgage or asset backed bonds. For our discussion, we will focus on the mortgage backed securities. The Mortgage Backed Securities (MBS) can be further classified into two broad categories by their defining attributes. The MBS security is backed by a pool of securities which can be residential-mortgage backed or commercial-mortgage backed. The Fig. 1 is a very high level of how an investor or an institution can invest into a Mortgage Backed Security (MBS) depending on their risk appetite.

We will discuss the structure of a mortgage pools which is relevant and in-scope for

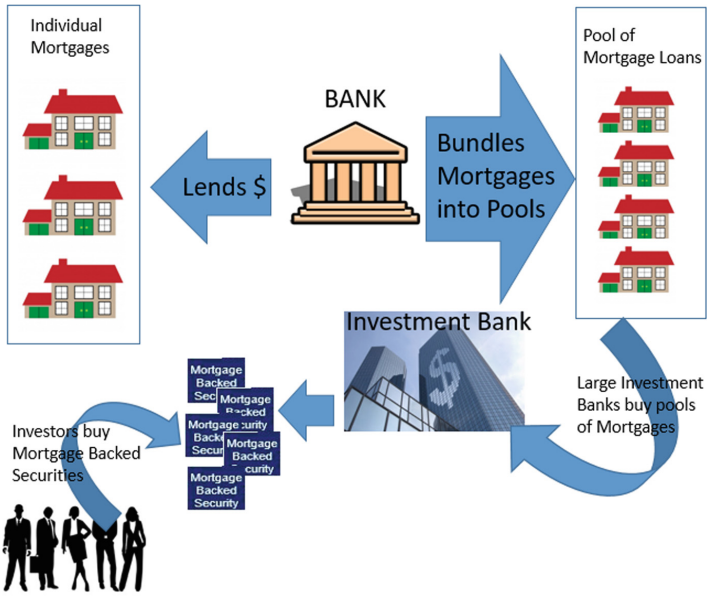

Fig. 1. Mortgage backed securities

this paper. The mortgage is a loan secured by the collateral of some specified real estate property which obliges the borrower to make pre agreed periodic payments. The lender usually banks, may require mortgage insurance to guarantee the fulfillment of the borrower's obligation. Some of the borrowers can be qualified for mortgage insurance which is guaranteed by one of the three U.S. government agencies, Federal Housing Administration (FHA), the Veteran's Administration (VA), and the Rural Housing Services (RHS). There are several types of mortgage designs used throughout the industry. A mortgage design is based on specification of interest rate, term of the loan and the manner in which the borrowed money is repaid to the lender. A pool of mortgage loans put together as a single asset is called a mortgage-pass through security. The cash flow of a mortgage pass through security depends on the cash flow of the underlying pool of mortgages. As each loan in a mortgage pass through security can have a different outstanding balance on the loan and different maturity it is customary to use a weighted average analytics for the overall pass through security. Figure 2 is a simple example of a pooled mortgage loan as a pass through security.

Furthermore, the MBS which are sold by Investment banks are divided into various tranches depending on the risk profile of the individual loans. These mortgage backed securities are very complex and the details are out of scope of this paper.

The Weighted Average Coupon (WAC) and Weighted Average Maturity (WAM) for the above mortgage pool is calculated as

Loan	Outstanding mortgage balance	Weight in pool	Mortgage rate	Months remaining
1	$125,000	22.12%	7.50%	275
2	$85,000	15.04%	7.20%	260
3	$175,000	30.97%	7.00%	290
4	$110,000	19.47%	7.80%	285
5	$70,000	12.39%	6.90%	270
Total	$565,000	100.00%	7.28%	279

Fig. 2. MBS pools, loans and their analytics

$$\text{WAC} = 0.2212 \times (7.5\%) + 0.1504 \times (7.2\%) + 0.3097 \times (7.0\%) + 0.1947 \times (7.8\%) + 0.1239 \times (6.90\%) = 7.28\%$$

$$\text{WAM} = 0.2212 \times (275) + 0.1504 \times (260) + 0.3097 \times (290) + 0.1947 \times (285) + 0.1239 \times (270) = 279 \text{ months (rounded)}$$

These are just 2 analytical measures but for a mortgage loan there are several of these measures which are important and which are calculated from the raw data set for investment management. All the different types (VA, FHA, RHS etc.) [1] are important and their % exposure in a loan is critical for portfolio manager's decision. This data is huge and some of the pools can be of thousands of loans. There are several calculations done on various dimensions (range bound). E.g. Showing the WAC over 0–2, 2–5, 5–7, 7–10 and above 10 may be one set of range and the other can be 0–3, 3–5, 5–7, 7–10 and above 10. Another example of a measure is Conditional Prepayment Rate (CPR). CPR is a sub calculation on Single Month Mortality Rate (SMMR) which is calculated for each month of the mortgage and use that SMMR to calculate the CPR for a specific month. The Eq. (2) shows the calculation for one month CPR. SMM is calculated with as shown in Eq. (1). Some of the mortgage calculations are recursive in nature.

$$\mathbf{SMM}(\mathbf{t}) = \text{Prepayment in month (t)}/(\text{beginning mortgage balance for month (t)} - \text{scheduled principal payment in month (t)}) \tag{1}$$

$$\mathbf{CPR}(\mathbf{1}) = 1 - (1 - \text{SMM}(1))12 \tag{2}$$

In U.S., there are three major types of passthrough securities are guaranteed by agencies created by Congress. This is done to increase the supply of capital to the residential mortgage market. Those agencies are the Government National Mortgage Association (Ginnie Mae- GN), the Federal Home Loan Mortgage Corporate (Freddie Mac- FH) and the Federal National Mortgage Association (Fannie Mae- FN).

The current paper is not focused on the solution which is provide to eliminate the EXCEL based solution but focus on scalability and flexibility by migrating the solution it to Cloud Computing environment. The migration of our existing SQL Solution

(replacing EXCEL based application) to Cloud Computing will give us savings in infrastructure, software costs. By solving the space constraints on data storage brings trend analysis in decision making [6].

3 Problem Statement, Related Work and Proposed Solution

The problem of having a flexible and scalable mortgage analytics from the huge data set by using EXCEL as calculation tool is very challenging and prone to three major issues. The EXCEL solution Extract Load and Transform (ETL) the raw agency mortgage files is not sustainable or practical and is very inefficient for decision making. Each file can contain millions of tuples or rows. The second issue of EXCEL as a calculation tool limits the ability of flexibility for any new metric calculations which are frequently needed on either ad-hoc basis or on a permanent basis. The modifications of EXCEL macros to accommodate any new calculations is difficult in our current version of EXCEL due to the complexity of code and lack of any version control or testing environment, opening a huge Operational Risk for the organization. The third constraint is the storage of historical data for trend analysis or data mining. The issues we have in our EXCEL version can be categorized into three main problem segments. (1) ETL is only possible for one deal at a time and is time consuming. (2) User defined calculations are not possible. (3) Trend and Data Mining capability is not available. Solving these three issues will increase the productivity of our investment teams giving more time for analysis rather than working on data collection and code manipulation.

To address the above mentioned EXCEL solution issues, our research focused on the existing academic and industry research on these issues of huge data stores and ETL tools to address our problem (1) and flexible calculation frameworks and data mining tools to address (2, 3) issues. There is a lot of academic research in terms of huge data store/ETL [16] and data mining tools.

The area we found is challenging is the flexible user defined calculations and running it on a distributed and on elastic data lakes and migration of SQL databases to cloud environment [10]. The mortgage pools are made up of individual loans. The data size of these pools can be viewed as Big Data. The main characteristics of Big Data is defined by the main three pillars by which we categorize the underlying data set. They are called 3Vs. [7].

The Fig. 3 shows 3Vs of Big Data in our study are Volume (Terabytes of data) in raw format with the Variety (Mixed data values) and Velocity (Changing daily) which makes it a best candidate for our study and implement our method of Capture, Transform, Calculate and Visualize (CTCV) to build a framework which can be leveraged for other investment data related decision systems.

The volume of the data is the number of tuples and attributes we have for each underlying pool. The distinct values or the domain values of each individual attribute can be a vast range of values which can be classified as variety of data set and the data changes from time to time and for our study we take a month over month change to

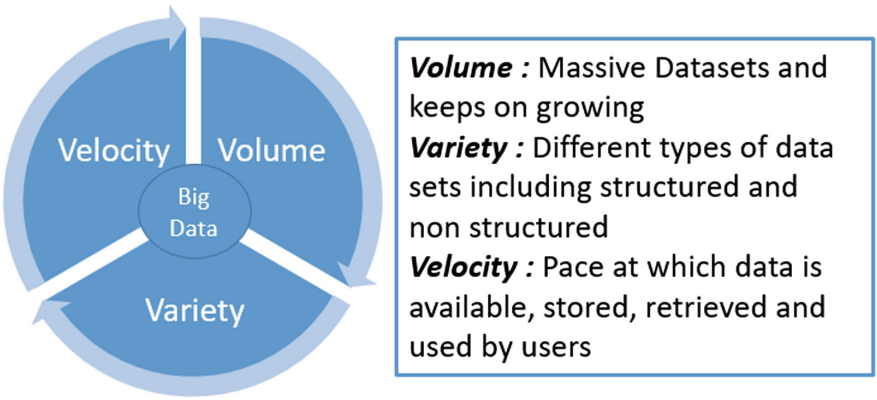

Fig. 3. Big Data categorization diagram

show the velocity is frequency and in our case it is monthly. There are several academic research papers on the Big Data and how we can leverage the existing technologies to source and store the data in a meaningful way. Big data analysis systems are very important to organizations because it enables them to collate, store, manage, and transform vast amounts data at the given instance, in a dynamically changing environment to gain the meaningful insights. The evolution of data management and knowledge systems [18] grow as necessity than nice to have. The initial sources are our raw data coming from various sub data base systems, flat files or any other source of information. In early 1970s the advent of relational data base systems (RDBMS) took the data store, retrieval to another level moving the needle from file based systems to relational set dependent systems. Codd's [14] paper on relational database design changed the way we process the data.

In academia as well as in industry there is a major research work carried out in terms of supporting large files coming from various source systems. During 2002 and 2003 Google came up with their proprietary file system Google File System (GFS) [4] which led the way to several other research groups to come up with their architecture to support large file systems. Google also published their MapReduce [5] programming model which can process terabytes of data on thousands of machines. MapReduce program was distributed over large clusters of commodity machines. During this time, Apache open-source developed Hadoop architecture and called it as Hadoop [17] Distributed File Systems (HDFS) and introduced their MapReduce programming model. The MapReduce architecture uses a map function specified by users that processes key-value pair to generate a set of intermediate key-value pairs, and a reduce function that merges all the intermediate values with the intermediate key [4]. Even though this is out of scope for our current paper, we are laying the architectural foundation for future work on processing our large data sets of mortgage pool information on a Cloud Computing environment using these distributed techniques. We keep our implementation open for our future needs.

As part of our research we evaluated several industry products in the Big Data and Cloud Computing space shown in the Fig. 4 including Oracle Cloud Platform,

Microsoft Cloud Technology (AZURE) and Amazon Web Services (AWS). All these vendors offer various cloud platform products. The best suited for our needs is Microsoft platform as our **Phase I** (Elimination of EXEL based solution) ***proposed and implemented solution is on a Standalone Microsoft SQL server***. This gives us a head start and saves resource time in learning other technology platforms which could be impact our business processes. Phase I used a lot of advanced normalization techniques [11] to improve the retrieval of information faster than traditional frontend driven controls in EXCEL. We utilized the RDBM concepts and techniques which helped us in solving several key issues in building BI dashboard [12] replacing EXCEL Solution. The other widely used Key-Value pair indexing [13] technique is used to build a distinct values list which serves the purpose of saving hard disk space and enables us to use lot of SQL In-Memory operations. Compressed Index Architecture of Microsoft did proved support in reducing our space constraints [3]. The proposed solution of migrating the standalone SQL Database solution to Azure Cloud Computing environment will provide the next stage for our "Financial Calculation Framework on Cloud Computing Environment".

As part of our discovery phase (research) we evaluated several cloud technologies provided by vendors.

The Fig. 4 shows various vendors we evaluated in search of a suitable vendor for our solution. The Microsoft Azure Cloud platform is open and flexible cloud platform providing ability to build, deploy, manage applications across global network with various tools and programing language support to enhance an organizations ability to

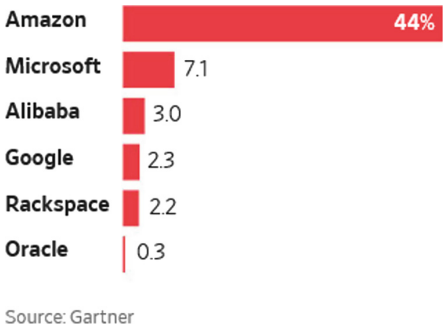

Fig. 4. Industry players in cloud computing space

grow with technology outsource to minimize costs and maximize profits by focusing on the investment performance to grow in market place. With in-house expertise we decided to migrate our standalone on-premises phase I solution to Microsoft Azure Cloud Computing Environment.

The "pay as you use" model of Cloud Services are key for the flexibility and expandability of any organization's growth. The main architectural components of any Cloud Service [8, 19] provider can be classified into three major categories. They are **SaaS** (Software as a Service) is centrally hosted and managed for the end customer and

usually based on a multitenant architecture. Example for SaaS can be Office 365 or Google Documents on Cloud. **PaaS** (Platform as a Service) is another service provided by the cloud service providers. In this architecture the customer will provide the application and the Cloud provider will provide the platform. **IaaS** (Infrastructure as a Service) is provided by vendor by running and managing server farms, running virtualization software, enabling the customer organization to create VMs (Virtual Machines) that run vendor's infrastructure.

4 Proposed Solution

After going through the pros and cons of Cloud Computing and the flexibility of the services provided by various Cloud vendors, we decided on using Microsoft Azure Cloud as our platform. We propose the solution in two phases. **Phase I** is building a standalone SQL DB solution to implement ETL and flexible calculation engine to calculate various analytical measures on our Big Data. This is not our focus of our current paper. The **Phase II** which is migration of the phase I solution is our focus.

The proposed Phase II or CTCV solution at high level, will take the existing installation of standalone in-house or on-premises solution and port it to Cloud Computing platform with minimal disruption to our business processes. The Fig. 5 gives a high level process flow of the events of our proposed implementation of CTCV.

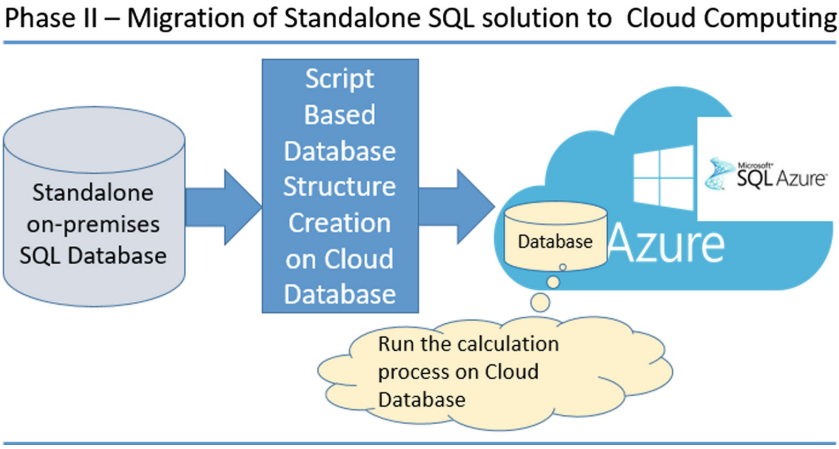

Fig. 5. Proposed solution architecture

Our preferred method is to take the SQL DB Script to build and implement the Cloud Instance rather than restoring the standalone database backup. Our approach of "Script Based" will have initial benefit of going through the objects where there will be changes needed to fit into the Cloud Computing framework. This shortens the implementation time and produces a cleaner and leaner version of the database schema.

With our script based framework it makes it more manageable by less sophisticated SQL users to migrate. The Other method of migrating the existing standalone databases with the backup sets which can create several issues (user access, roles, Active Directory groups (AD Groups) etc.) which a non-technical user can't be able to handle. In a "Backup Based Model" the testing of end-to-end may not be done till later phase. In a script based migration model we were able to test and modify the components needed as we move from one object group to another object group.

Our solution is easy to implement by any team or individual with some knowledge of SQL. With the simple steps which we propose will lead the organization to migrate to a more flexible and scalable data base solution in an organization. The flexibility and scalability comes from the fundamental design of a cloud architecture of "elastic" nature of the Cloud Computing services we discussed in our Cloud Architecture section (Related Work). The elastic nature of the Cloud Database Solutions enables our existing database to grown as the Bid Data grows as time passes. In the next section we will layout the actual implementation of our Mortgage Loan data with a sub set of information as part of our Phased implementation setup.

5 Implementation of Proposed Solution

Our proposed solution of migration of our Phase I solution (Standalone SQL Database) to Cloud Computing environment is done in two parts. Pre-migration and Post-migration. These two will set stage for a better and stable migration process.

Pre-migration: The first step is to verify the subscription of the Microsoft Azure Account with the institutional license management team and if not we need to obtain the needed subscription. Usually depending on the usage the organization purchases the subscription level. Once we have the proper license set up, check the access to the Azure portal (http://portal.azure.com). In the Fig. 6 we show the general layout of the Azure Portal which can be maintained by the subscription we have.

Our implementation is on a Windows Azure MSDN – Visual Studio Premium which gives us the ability to build elastic database servers with a 250 GB space. This is a called Standard S3: 100 Database Transaction Units (DTUs). We are using this space to prove the proposed architecture solution will work and can be implemented to bigger database needs. Our test data is for a sub set of Mortgage Agency Pools. The data set we have is for Agency FN and currently we have 12 raw files with 14 million total rows.

Proposed Implementation Steps: To implement our solution of Script Based, we first took the current implementation database (on-premises) schema script. These DDL scripts are saved as "Object Categories". E.g. For Tables, we called as AZURE – Table Script and for the Store Procedures are called as AZURE – Procedures. It is up to the individual team to decide the naming convention. This step gives the implementation team a change to modify any non-cloud scripts or rework the non-cloud scripts before implementing on the Cloud instance. One example of "script modification" is removing any cross-database-queries. Next step is to create a proper database server and database on the Azure Cloud. This can be done with simple steps of by going

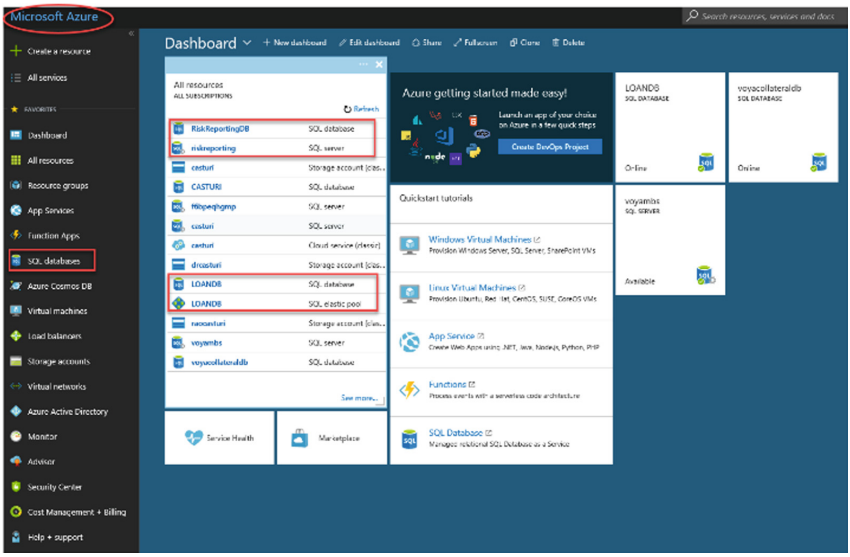

Fig. 6. AZURE set up of our MBS cloud database

through the Microsoft Documentation [9]. Once we have the Azure Cloud SQL Server and Database, the next step is to run the DDL scripts which we saved as "Object Category Scripts" our current implementation. If the SQL DDLs are standard, then we should not see any issues. We did encounter few issues which are rectified during the migration. The Azure SQL code won't let us reference the <database>.<owner>.<object>. The <database or Schema name> is not permitted in the Azure syntax of referencing the database objects. This will pose an issue with migration if we have linked servers or queries going across multiple databases. The newer version of Azure SQL has a fix.

Create an "External Data Source" as shown in the Fig. 7 uses the architecture in the Azure SQL Database under the External Resources and use that in DML when writing the standard query referencing the remote SQL table of SQL Server instance in

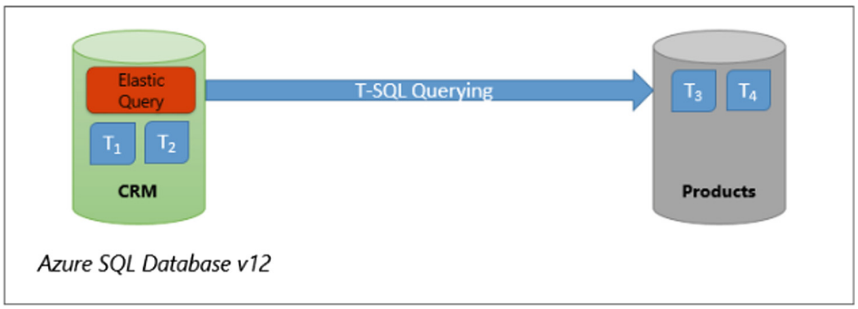

Fig. 7. Cross query set up schema.

Azure SQL installation. The Figs. 8 and 9 show the External Data Source Set up and how to reference and execute the cross database queries if needed.

Post Migration: Till this point of implementation, we used the Azure portal. Now it is time for connecting to the Azure Cloud via other client tools. Usually in the organi-

```
CREATE EXTERNAL DATA SOURCE RemoteReferenceData
WITH
(
        TYPE=RDBMS,
        LOCATION='myserver.database.windows.net',
        DATABASE_NAME='ReferenceData',
        CREDENTIAL= SqlUser
);
```

Fig. 8. Creating external data source in Azure [9]

```
CREATE EXTERNAL TABLE [dbo].[zipcode](
        [zc_id] int NOT NULL,
        [zc_cityname] nvarchar(256) NULL,
        [zc_zipcode] nvarchar(20) NOT NULL,
        [zc_country] nvarchar(5) NOT NULL
)
WITH
(
        DATA_SOURCE = RemoteReferenceData
);
```

Fig. 9. Data source concept in Azure SQL database [9]

zation, the data base professional (Database Administrators, Developers, Designers) use several client tools to connect to the SQL Database instances to do their regular tasks. The Azure SQL is no different and we would like to show the easy way to

connect to the already created Azure Cloud SQL database. For our project we used the Microsoft SQL Server Management Studio (MSSQLMS) as our client tool to connect to the Azure Database instance. While creation of the Azure SQL Server Azure will create a unique server name to reference the instance. The Fig. 10 shows the Azure SQL Server and also the Azure SQL Database with the names we provided for referencing them through our implementation. This is a big step as creating the database on Azure depending on the subscription level and we want to make sure we have enough service level contract in place to proceed.

Once the database schema is set up with the needed external sources, we then populate the mapping table data needed but importing the data from files or current

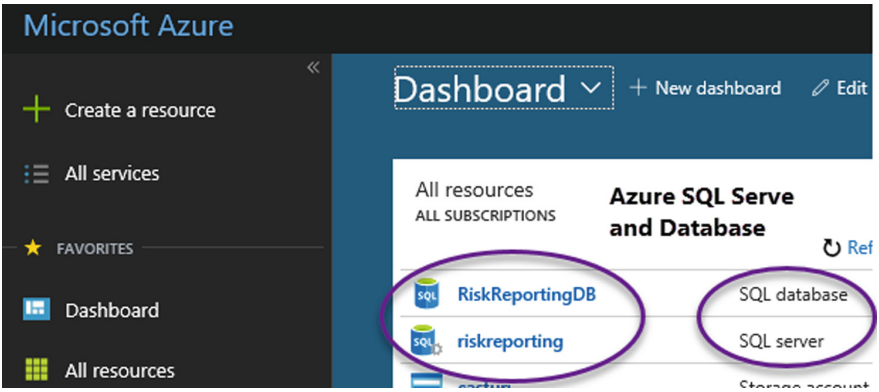

Fig. 10. Azure data dashboard

standalone SQL Database. For our instance we named our SQL Server as "riskreporting" and the instance can be referenced as "riskreporting.windows.net".

Now to connect to the Server and to access the database there is one more step involved which is done through the Azure Portal. This is called setting up the client machine firewall to access the database server. The Fig. 11 shows the Firewall setting.

Usually if we are going to implement the Active Directory Group (AD Group) access they can be done through centralize IT security function but for our proposal we are maintaining the access control. The other reason is to have data secured and protected from other non-access individuals even though they are on our AD Group. Once the firewall IP address is added to the Azure SQL data server we can now connect to the database via the MSSQLMS. Figure 12 shows the screen where we give the server credentials, in our case "riskreporting.database.windows.net" and the log in credentials which were created during the process of setting up the Azure SQL Server and database.

We tested both (elastic, non-elastic databases)the installs with our script based approach to make sure we test them as part of our solution. We can now connect to the

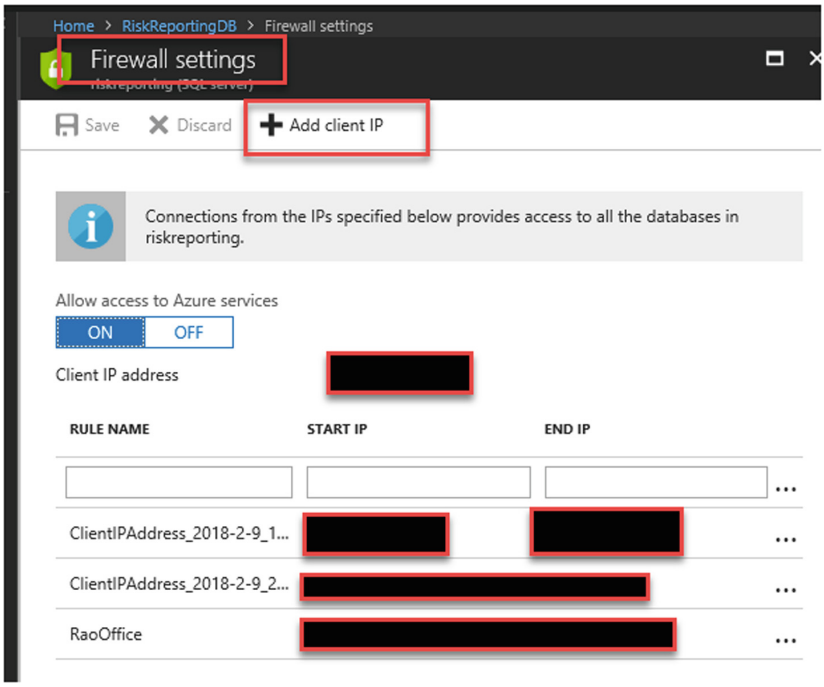

Fig. 11. The Azure firewall setting to allow the client access

Azure Cloud Database using the client tools by referencing the database name **riskreporting.database.windows.net**. This is shown in the Fig. 12 with the users credentials to be passed to have a the Microsoft SQL Query Editor to show the objects and the query panel to be available for the users.

Our Azure Database instance is RiskReportDB. The Fig. 12 shows the connection to the Azure SQL Server and the database on the server to which we have access. We set up 2 resource pools one with elastic (voyacollateraldb) and one without elastic database (RiskReportDB). This set up is to verify the elastic nature of the database which we need for our implementation to test the multiple month storage which solved our historical nature of the data store.

Database Transaction Unit **(DTU)** is a key measure for a single Azure SQL database at a specific performance level with a service tier. DTU is a blended measure of CPU, Memory, I/O (data and transaction log I/O). Once thing to note is the resources used by our workload do not impact the resources available to other SQL databases in the Azure Cloud and the resources used by other workloads do not impact the resources available to our SQL database. The DTU is a bounding box with CPU, IO and Memory as boundaries. The Fig. 13 shows the DTU box.

Once we have the database connection ready and the objects ready on Azure Database, we now can load the data via SSIS or any other standard ETL data load

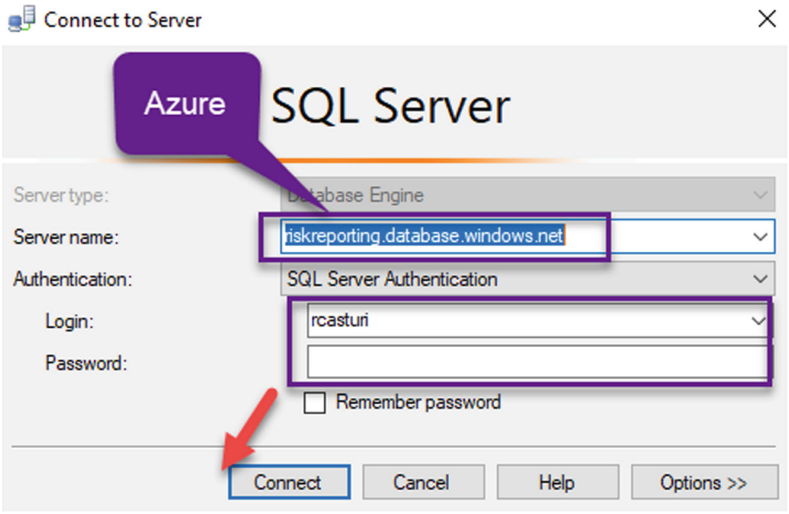

Fig. 12. Azure database connection via SQL client

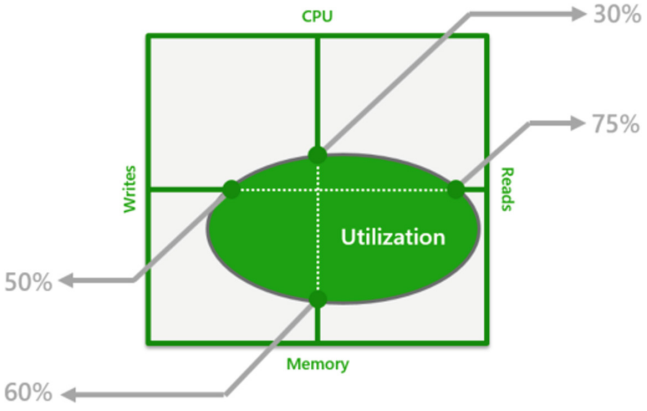

Fig. 13. Monitoring Azure database workload utilization within bounding box

methods. For our testing, we used the DTS Wizard to load the data for the users set up tables and for the factor data. With this step we are ready with all the needed base data set for the actual analytical calculations to be performed on our newly created Azure Database. We can run the calculation engine via client tool or via the Azure Portal. To run any SQL Queries on Azure portal we use the portal query panel provided by Azure SQL data server.

6 Results

We ran our user defined Analytical Aggregation Rules on the newly set up Azure SQL Cloud Computing environment. There were no errors. The final calculations were verified with the standalone SQL Database implementation which is our Phase I production. The verification is done by comparing the results from on-premises version to the Azure SQL Version. The approach is to take the calculations of one deal and compare it with the results produced by Azure implemented solution's calculation numbers. There are no deviations or errors in our newly implemented Azure SQL implementation (Phase II). The subscription we currently have is a very close match to the on-premises so we did not see the any significant improvement in performance. As we noted the higher subscription for more DTUs the better the performance. This is an advantage of utilizing Azure Cloud Computing services as we can increase our DTU subscription depending on our necessity. These are conservative estimates to give us ample room for migration and re-work if needed. As shown in the Fig. 14 we can see the spikes in the DTU usage which indicates the CPU on the Azure Server is active and processes the users requests.

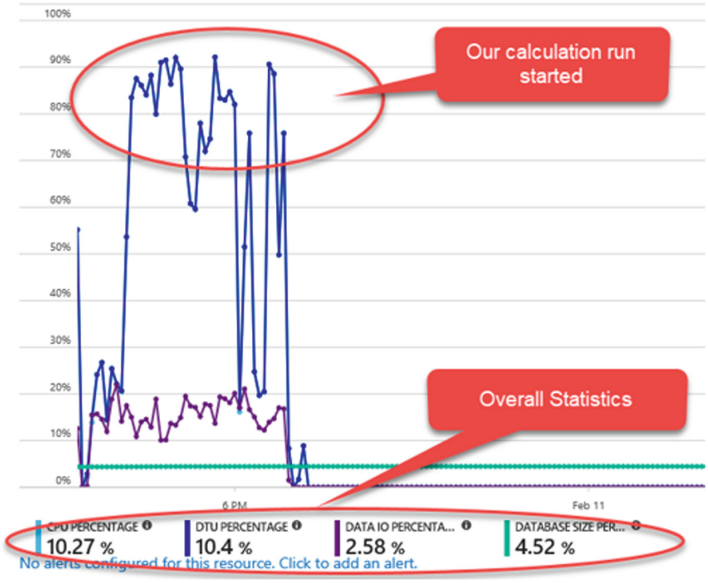

Fig. 14. Cloud computing calculation time chart

Another observation we did depending on the rating category by services without any business disruption. Here we applied ratings for several categories which are important for our organization. We recorded the platform (On-premises, Azure) suitable for our organization for future needs and plotted the rating of each category and assigned an appropriate rating between 0 and 5. The results of the rating based scale

will be discussed in our next section of conclusion as part of the comparison of the on-premises and Azure Cloud Computing Architecture. Figure 15 shows the plot by category and rating for each category.

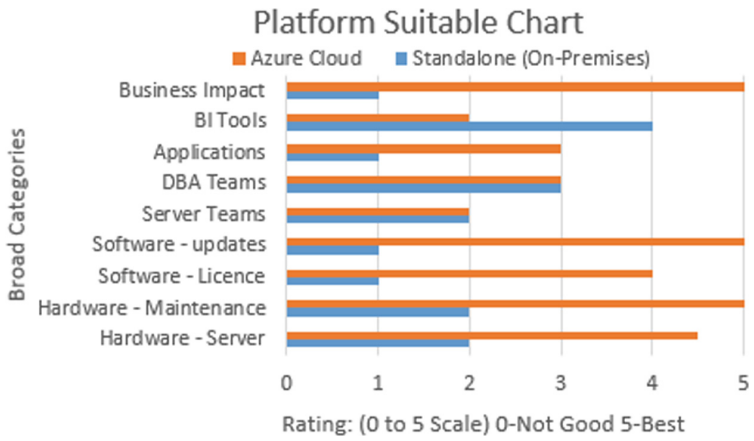

Fig. 15. Comparison (In-premises vs cloud architecture)

The higher the ranking the it is better suited product for our usage. As an example, if we take "Business Impact" category and compare the score of 5 for Azure vs. 1 for Standalone it shows that we can implement Azure solution won't impact business as it scores 5 whereas the standalone solution can impact business if there is any migration requests are made to upgrade or add additional infrastructure resources. Whereas in BI tool category, the standalone instance has more flexibility and less expensive compared to Azure solution. This gap can be narrowed as more research and reporting frameworks migrate to Cloud Computing environment in the future.

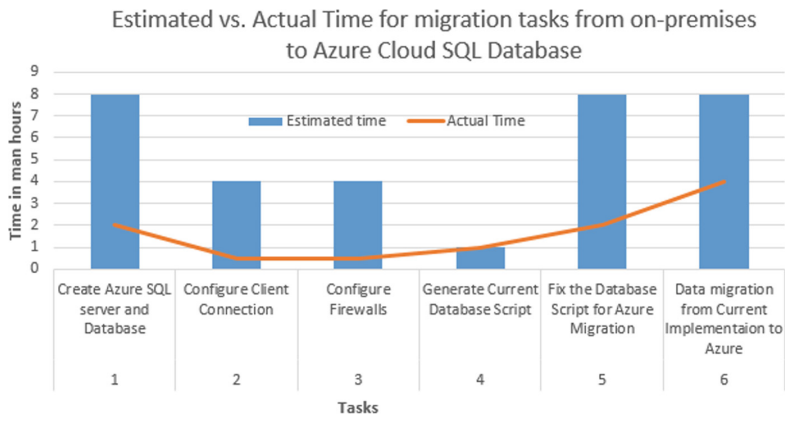

Fig. 16. Time chart (Expected vs actual implemention)

7 Conclusion

The results of migration from a standalone or on-premises SQL Database to Cloud was a successful and this is due to the preparation of the pre-implementation preparation we took before actually implementing the DDLs on Azure SQL platform. The initial estimates of our migration was very conservative giving us enough room to implement any work around for migration scripts to Azure SQL database. We plotted the initial estimated time for each task and the actual time taken for the task to implement. The graph in Fig. 16 shows the solid bars are estimated and the line showing actual time we took to implement. The shorter times for actual implementation can be largely contributed to the preparation and project planning going through various scenarios and studying several case studies which are available in industry and also on Microsoft's knowledge base.

This actually helped a lot in preparing the perfect "scripts" to upload and run on Azure SQL Database. Usually the rework is in making the scripts run without errors and that we were able to accomplish early in our implementation phase. We called this as pre-implementation task.

The Infrastructure as a Service (IaaS) [10] architecture promised by Microsoft is actually worked well for us. Due to the planning and execution the implementation by script based approach, we were able to achieve building the foundation of our migration tool kit to manage Big Data Management. The growth of asset under management AUM depends on the flexible and scalable enterprise architecture and we believe Microsoft Azure Cloud Computing platform solutions will enable to travel on a growth trajectory. The contribution of this paper is not just for any investment company but can apply to any data driven business in industry or in academia.

With all the above mentioned positive outcomes, we conclude the migration of standalone SQL database to Azure Cloud Computing environment will increase our investment portfolio manager's productivity, cut down the infrastructure costs of buying hardware (servers) and maintaining the software patches, building performance tuning tools in-house. The hard dollar saving can be put to work and can show profitability. The overall business goal of flexibility, scalability is accomplished by migration to cloud computational model. The growth of asset under management AUM depends on the flexible and scalable enterprise architecture and we believe Microsoft Azure Cloud Computing platform solutions will enable to travel on a growth trajectory.

During our migration process we noted there are few things we can share with other researchers who are in process of migrating from on-premises databases to cloud computing environment. If an organization decides to utilize the Cloud Computing Platform, there are few things should happen before they can start doing any production migrations. The first and foremost is to justify the need for cloud computing. This won't be suitable for a small company with fewer resources and there no requirement for computational needs to support the product or functions. If the organization's data can be managed and maintained on one or 2 SQL servers with one or two Database Administrators (DBA's) then cloud solution may not be an optimal solution. Building the infrastructure around the cloud will defeat the purpose of simplification and cost saving for the organization if the data is contained and there is no need for future

growth. The second main point we recommend is to evaluate the subscription services provided by the Cloud vendors. There are several options from which an organization can select the products depending on the need seen by the business. It is very important to able to assess the need as building a solution in a lower DTU subscription and moving it to a higher DTU is a bit challenging. This can impact overall enterprise contract negotiations which can delay or impact the cost of project. The third point is on deciding elastic database architecture. Having the ability to grow and able to utilize multiple databases in cloud is better than having multiple standalone database servers and link them through a distributed database architecture. The other options we would like to put it out for the researchers are Microsoft Azure Data Lakes and Data Warehouses. These can be used and enhance the company's productivity and reduce the dependency on standalone on-premises infrastructure saving cost (time and resources) in future for the organization.

The contribution of this paper is not just for any investment company but can apply to any data driven business in industry or in academia. The tool kit we proposed which includes the script based implementation framework along with the set-up of the client and development tools will make any migration seamless and very little downtime for an organization.

8 Future Work

As we mentioned earlier in the paper, this project implementation is a multiphase implementation designing each phase to be the launch pad for the next phase. From EXCEL version to the Standalone SQL Solution itself is a huge project which included flexible user calculations. During that design phase we did keep our next phase of implementing the standalone solution on Cloud Computing environment as that technology is evolving and we do have need to embrace that technology as the hardware, software cost is getting higher for a standalone SQL Solutions. With the implementation of the migration project, the future is to build the analytical trending models on the Cloud Computing platform to give our portfolio manager the cutting edge technology for the day to day investment decisions.

Our implementation added value to our organization and paved the path for other teams to migrate from standalone database instances to go in to next stage of Cloud Computing. Our paper and our successfully implementation demonstrated that script based implementation is worthwhile undertaking in migrating several of our on-premises databases to Azure Cloud Computing environment. The Microsoft tools [19] we mentioned and proposed will help mid-sized company to migrate their existing on-premises databases to cloud based architecture using the simple procedures mentioned in our paper. The methods are be applied to any database driven applications in any organization. The solution is flexible and scalable and in future we are trying to make is a standard migration tool for our organization. The tool kit we proposed which includes the script based implementation framework along with the set-up of the client and development tools will make any migration seamless and very little downtime for an organization.

The other important area of our future research is on security and privacy [20] of the Cloud Computing Architecture. This currently is a concern to many financial organizations. The main areas of focus will be on the rules and controls around "Proprietary and Intellectual Property" and cyber intrusion of company financial data. We are currently undertaking this research and will bring out the current issues and possible solutions in building a framework for any financial institution to be able to utilize the cloud computing environment.

The other area we are focusing our research is in implement a full stack BI and Data Mining [15] framework on our Azure Cloud Computing Solution.

References

1. Fabozzi, F.: Fixed Income Analysis. CFA Institute Investment Series, 2nd edn. Wiley, Hoboken (2007)
2. Fabozzi, F.: The Hand book of Fixed Income Securities, 7th edn. McGraw-Hill, New York (2005)
3. Ozur, M., Tuna, H., Coffin, C., Sampaio, T.: Azure Virtual Datacenter. Microsoft, November 2017
4. Ghemawat, S., Gobioff, H., Leug, S.-T.: The Google file system. In: SOSP 2003, Bolton Landing, New York, USA, 19–22 October 2003
5. Gemayel, N.: Analyzing Google file system and hadoop distributed file system. Research Journal of Information Technology **8**(3), 66–74 (2016)
6. Weber, N.: Big Data: can we prevent the next financial crisis? King's College London Economics & Finance Society (EFS)
7. Eaton, C., Deroos, D., Deutsch, T., Lapis, G., Zikopoulos, P.: Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill, New York (2011). ISBN 978-0-07-179053-6
8. Microsoft Azure Documentation: Overview of Azure Data Lake Store, Microsoft
9. Microsoft Azure Documentation: Create an Azure SQL Database in Azure portal and Linked Databases Microsoft
10. Microsoft Azure Documentation: Load data from flat files into Azure SQL database. Microsoft
11. Elmasri, R., Navathe, S.: Fundamentals of Data Base Systems, 7th edn. Pearson, London (2015)
12. Esling, P., Agon, C.: Time series data mining. ACM Comput. Surv. **45**, Article no. 12 (2012)
13. Zaharia, M., Wendell, P., Konwinski, A., Karau, H.: Working with key/value Pairs. In: Learning Spark. O'Relly Media, Inc., February 2015
14. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
15. Grossman, R., Gu, Y.: Data mining using high performance data clouds: experimental studies using sector and sphere. In: AMC KDD 2008, Las Vegas, Nevada, USA, 24–27 August 2008
16. Apache Hadoop. http://hadoop.apache.org/
17. White, T.: Hadoop The Definitive Guide, 3rd edn. O'Reilly Media Inc, Sebastopol (2012)
18. Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2012)
19. Microsoft Press: Cloud Application Architecture Guide
20. Sobati Moghadam, S., Darmont, J., Gavin, G.: Enforcing privacy in cloud databases. In: Bellatreche, L., Chakravarthy, S. (eds.) DaWaK 2017. LNCS, vol. 10440, pp. 53–73. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64283-3_5