

Time Performance Evaluation for Batch Processing in Salesforce.com Cloud Computing

¹Awfa Hasan Dakheel and ²Alaa Shawqi Jaber

¹College of Basic Education, University of Babylon, Hillah, Iraq

²College of Information Technology, University of Babylon, Hillah Iraq

Abstract: Salesforce.com is a web-based enterprise platform offers to developers an environment for many objectives such as development, testing and training. The ability to import data such as accounts/contacts, leads, solutions and custom objects to Salesforce are important and usually it may be required, depending on the nature of process adoption by Salesforce.com. The objective of this study is to lend some understanding on Salesforce.com in terms of calculating the time of invoking different volumes of batches which represents in a job that queued in Apex job queue to be executed. This study highlights the awareness of evaluating the time performance when need to execute a large amount of data on Salesforce.com platform. The study endorses that the awareness of the time performance in execution different volumes of Apex jobs will aids to estimate the consuming time when need to process batches exceed 500. Moreover, the study probed into Model View Controller (MVC) for Salesforce.com environment to fulfil the aforementioned requirement.

Key words: Salesforce.com, salesforce, Apex batch processing, Apex job, standard objects, MVC, cloud, computing

INTRODUCTION

Broadly speaking, technology has had a profound effect on almost every facet of life. Salesforce.com is a web-based enterprise platform that can be getting into from anywhere, anytime and on any device as long as there is an availability to connect to the internet (Yu, 2015). It is a cloud application, like Gmail or Yahoo e-mail but it takes this a step further because it allows hardware, the operating system and the database to share among Salesforce customers within the same “instance” (SD., 2016a).

Now a days, a developer in Salesforce.com can create multiple and long-running processes by utilising batch Apex on the Force.com platform. In another word, a user may build a system for archiving that seeking for old records which past a certain date then archiving them. Moreover, a developer could update all accounts and opportunities responded to custom criteria by building a data cleansing operation (Salesforce, 2016).

Managing time in every sector of our life is a crucial issue. The consuming time for processing of part or multipart of batches in Salesforce’s platform has a magnificent consideration. It’s give a perceptive understanding in handling time when need to process bunch of batches in the Salesforce.com platform that exceed the permitted uploaded batches limit. Identifying a custom batch size is affordable in Salesforce’s

platform. For a partial processing, the developer can detach batch transactions. Moreover, this has a benefit when a failure in rolled forward for one batch is happened; the other batches will not disturb and not rolled back if they are successfully processed. However, every partial or whole processing for batches will receipt varied execution time. For this reason, this study aims at shedding some light on implementing batch processing using Apex language and calculating the time of execution for each batch request. The developer will be aware and considered of the spent time in practising the processing of batches in range from 1-500. To fulfil the aforementioned requirement, it is critical to take an overview of the fundamentals of the Salesforce.co and its platform.

Theoretical background: In this study, we will give an overview about Salesforce.com and describe the architecture of Model View Controller (MVC) in cloud.

What is salesforce: Salesforce.com is complete software offering on the cloud. Initiated in 1999 by Marc Benioff (Oracle), Parker Harris (Left Coast Software), Dave Moellenhoff and Frank Dominguez (both from Clarify), Salesforce.com began as a (Customer Relationship Management: CRM) product but it developed after the years into much more. High level products in Salesforce.com not limit to: sales, service and support, partner

relationship management, marketing, content, ideas, analytics and chatter but also by using apex and visualforce proprietary languages the developers allowed to write applications on the Force.com platform. These services by Salesforce.com provided “in the cloud”.

In order to secure the platform with the possibility of reducing the access of unauthorized people to others computers, Salesforce has established a number of mechanisms. Different foundations have a number of accessing methods to the application of the Salesforce. In fact the access can be granted by the interface of a user for example by using web internet browser, Apex data loader (client application) or an Integrated Application (API) as a client desktop (salesforce to be used with Outlook) or from application of a mobile (Goodey, 2011). Since, salesforce.com is a cloud application, it is essential to investigate the architecture of cloud computing. In general, according to Furuncu and Sogukpinar (2015), Selvakumar *et al.* (2013), cloud computing have three levels IaaS, PaaS and SaaS which are abbreviations for Infrastructure as a Service, Platform as a Service and Software as a Service, respectively. The three levels of cloud computing are illustrated in Fig. 1.

IaaS performs the lowest level it provides a physical infrastructure to renters and offers storage services or virtual servers. Foremost vendors are Amazon EC2, Amazon S3, Rackspace Cloud Servers and Flexiscale. PaaS lies between IaaS and SaaS. PaaS offers an advance platform on the cloud. It usually provides solutions for problems or helps to build solutions by constructing blocks. The representative companies in PaaS are Google’s Application Engine, Microsofts Azure and Salesforce.com. SaaS is the third level of cloud computing, it contains complete software offering on the cloud. The innovator in this field is Salesforce.com, provides its own customizable user interface and proprietary Apex programming language. Googles, Gmail and Microsofts hotmail, Google docs and Microsofts online version of office called BPOS (Business Productivity Online Standard Suite) are considered as online email providers and examples of SaaS.

Benefits of salesforce.com: According to Anderson (2013). Several behavioural, technological and managerial forces are dramatically and irrevocably changing the way that salespeople and sales managers understand, prepare for and accomplish their jobs. Field salespeople, operating out of mobile virtual offices are being empowered and becoming increasingly independent as they shift focus from selling to serving customers. Meanwhile, the sales

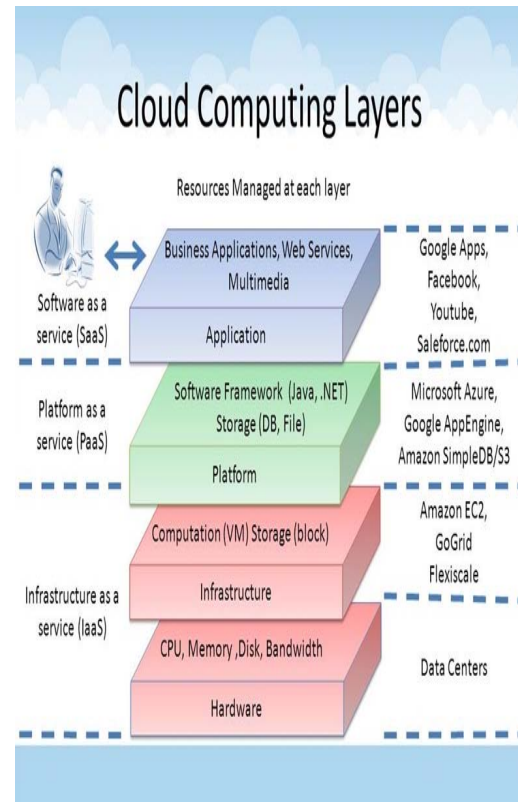


Fig. 1: Cloud computing layers (Anonymous, 2017)

manager’s job is evolving more toward that of channel manager overseeing a hybrid sales force serving customers in diverse.

Sales management represent three themes as the main keys in Salesforce. First one is how to think outside the box (innovation preparedness), the ability to do things in a different way and adopt change. Second one is by applying technology the broad spectrum of technological tools now available to sales managers and sales organizations. The third one is the leadership which is the ability to make things happen and bring the benefit for the organization and the customers related to it (Johnston and Marshall, 2016).

Salesforce.com was intended to act as a CRM application only. However, as the platform grew and became more robust, Salesforce opened up its platform to more customization where it offers to build any kind of applications beyond CRM. The following products are available in this platform (Yu, 2015):

- Sales cloud: this is used to automate the sales process
- Service cloud: this is used to deliver revolutionary customer service process

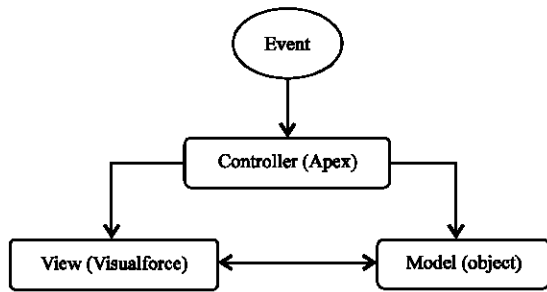


Fig. 2: MVC on sales force.com

- Marketing cloud: this is used to provide digital marketing automation
- Analytics cloud: this is used to deliver analytics for business users and analysts
- Force.com platform: this allows building own Enterprise custom apps that connect with customers, employees and partners

In summary, the advantages of using Salesforce.com compared to other on premise applications are as follows (Yu, 2015):

- Faster implementation schedule
- Lower maintenance cost, since you don't have to buy or support in-house servers, data centres and high-speed internet connections or hire any IT staff for this work
- It is scalable and robust
- Security and high performance
- Easily expandable functionality using prebuilt solutions from the AppExchange
- Access from desktops, laptops and mobile devices with internet connectivity
- Enjoy enterprise grade level application for small and medium businesses

MVC in the cloud: Model View Controller (MVC) is a prevalent architectural design pattern in modern software development. It promotes the separation of an application's logic, user-interface and data storage from one another. This architecture has risen in popularity as this isolation of data (the model), user-interface (the view) and logic (the controller) allows each component to be developed, tested and maintained independently. The Force.com platform's development architecture fits conveniently within the MVC pattern allowing development teams to take advantage of the separation of responsibilities. Figure 2 shows MVC on Salesforce.com.

Table 1: The main objects in salesforce.com (Yu, 2015)

Objects	Descriptions
Account	Users or entities involved in business relationship. Those may be business clients, partners or competitors
Contact	Store information about the people associated with our customers
Opportunity	Store information about sales interactions with or customers
Case	Description of the problem that a customers may have raised
Report	Analysis of data stored in standard or custom objects
Dashboard	Groups of charts or graphical data generated from reports

The model (salesforce objects): In salesforce.com objects are key ingredient. Storing data is the main function of objects. Like the tables in a database, objects contain several fields to store data. We can illustrate an object as a table, a field as a column in the table and a record as a row in the table. In general, every day, it is vital to manage thousands of records in the medium and the large organizations. This includes deleting, editing or adding these records when it is needed. The salesforce has provided what is called (Batch Apex) which is considered as a powerful concept that provides the ability to manage more numbers of records with the potential to manipulate these records by using specific syntax. Objects in salesforce have two kinds (Yu, 2015). Standard object and Custom object.

Standard objects are provided by Salesforce.com by default. Each standard object has its own uniqueness and the objects are named by their common uses. In this study the focus is limit to use the standard objects. Some of the main Salesforce.com objects are as follows in Table 1 Custom objects are specific objects created in organization to store data for specific business that does not fit into standard objects. Only a user with admin access is allowed to create custom objects. The maximum number of custom objects that can be created depends on the Salesforce edition (Salesforce, 2016).

Controller: The most complex area of the Force.com platform is largely regarded to be the Apex language and its implementation. Apex classes together with Apex Triggers form the controller in the MVC pattern they provide an interface to the model and expose their logic and functionality through the view.

Apex is a strongly typed, object-oriented language similar in syntax to Java or C# and (in certain areas) Oracle PL/SQL (SD., 2016b). It's including a "Data Manipulation Language (DML) calls such as INSERT, UPDATE and DELETE and a built-in DmlException handling. Also, it provides the conditional logic, various looping constructs, data retrieval and manipulation, transactional control, expression evaluation and error handling common to its class of programming languages.

Apex classes and Java classes have similarities. A class considers as template or blueprint to form objects. Apex class contains variables and methods. Variables can describe the status of an object, for example, name or type of the object. Member variables are defined as variables that associated with a class and members of it. To control behaviour methods are used, for example buying or selling an item. Method may include local variables with their declaration inside the method. Only method can use these local variables. Although, the variables of the class member identify the lineaments of an object, for example the name or the height whereas the local variables doesn't describe the class, however; it is used only by the method (Salesforce, 2016).

View: Force.com platform has a component-based framework which is called Visualforce. To build dynamic components and pages that are rendered as HTML and JavaScript at runtime, Visualforce uses an XML tag-based language.

By using visualforce, combination data from multiple objects, creation mashups with data from external web services and even overridden some of the logic and the behaviour found within standard Salesforce CRM application functions are all available. There are three elements of Visualforce (Goodey, 2011). Visualforce pages, user interface can be defined by using these pages. Visualforce components, Visualforce code can be found in a library of standard or custom-built sections. Visualforce page controllers, they are important to control the behaviour of Visualforce pages and can either be controlled by standard logic or can create custom logic to change or extend the standard Salesforce CRM behaviour.

The language includes a large, well-documented component library as well as the ability to create own components. Standard and custom components include a number of convenient functions familiar in most front-end languages as well as baked-in AJAX functionality. Additionally, we can include any amount of HTML and JavaScript in Visualforce pages allowing for the creation of sophisticated, modern user interfaces. These interfaces can also be context aware in that they render differently when viewed on different devices such as slate computers or mobile devices (Salesforce, 2016; SD., 2007).

MATERIALS AND METHODS

For the batch class implementation process, it is called or recalled one time for each batch of records. The default batch size is 200 records or one batch and there is

an ability to identify a custom batch size. Moreover, when execute batches; each batch can be detached or split. This has an advantage in which the successful processed batches will not roll back if one batch fails to process. Apex class are used to perform batch Apex in which it implements the Salesforce provided interface (Database.Batchable<sObject>) and then call the class programmatically. There are three execution methods for each batch of records starting with start method, then (execute), method and finish method (SD., 2016b). When start provides different number of batches, this will be processed by start method, it returns the list of records to be processed by calling the Database.QueryLocator (queue); this list to execute should override to the second step the execute method which is basically "insert" or "delete" with DML statement. After all batches are executed the finish method is called. Sending confirmation emails or execute post-processing operations will be performed in this method.

At the beginning of a batch Apex in the Database.Batchable interface, the "start" method will be used as follows.

Start method: Global (Database.QueryLocator| Iterable<sObject>) start (Database.BatchableContext bc){}. The start method is called a job that used to collect the records ranged (200-50000) records and passed to the interface method execute. This method returns either a Database.QueryLocator object or an Iterable that contains the records or objects being passed into the job.

Execute method: Global void execute (Database.Batchable Context BC, list<P>){} each batch of records passed to the method is called for the execute method. Use this method to do all required processing for each amount of data. The following actions will be performed:

- A reference to the Database.Batchable Context object
- A list of sObjects such as List<sObject> or a list of parameterized types
- In execution, the order of records batches is changing when received from the start method

Finish method: Global void finish (Database.Batchable Context BC){}. After all batches are executed, the finish method is called. With this method the confirmation emails will send or execute post-processing operations. To invoke a batch class, the following statement will be called. Figure 3 shows the flow diagram explains the paper method for batch processing in the database system.

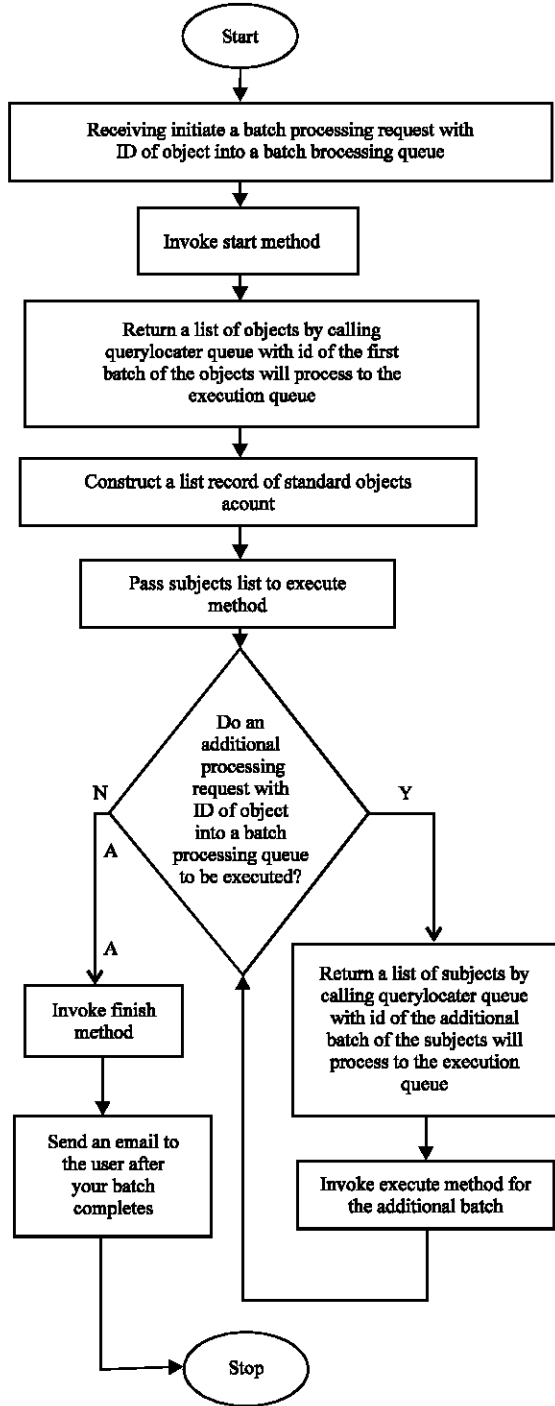


Fig. 3: Flow diagram of high level overview for apex batch processing

RESULTS AND DISCUSSION

In this study, the invoking was made to performed different number of batches ranged between (1-500). Since, the default batch size is 200 records, so this

Table 2: Apex jobs details

No. of records	Total batches	Sub-mitted time	Completion time	Execution time (min)
200	1	20:20	20:20	0
1000	5	20:21	20:21	0
5000	25	20:22	20:23	1
10000	50	20:38	20:43	5
25000	125	20:46	20:58	12
50000	250	21:00	21:30	30
100000	500	21:37	22:28	51

Database execute Batch (myBatchObject)

variation in the number of batches will highlight the differences in evaluating the time needed to execute each Apex job or batches.

According to Yu (2015), importing data can process up to 50,000 records at a time. If the volume of data is larger, divisions into many files should be made over the volume of data.

The invocation was executed over a proposed range of batches which is (1-500) to show the batches job execution status, the starting time and the completion time since the job had finished. Figure 4 indicates the proposed status that implemented in Salesforce.com’s platform.

Table 2 will describe the details of Apex jobs recorded in Fig. 4. It shows different times of execution in processing various number of Apex jobs. For example, the starting time to execute 50 batches was 20:38 and the completion time was 20:43, therefore, it took only 5 min to complete the process. Likewise when processed 250 batches, the starting time was 21:00 and the completion time was 21:30 as a result the consumed time for the execution is 30 min. The line chart in Fig. 5 shows the execution time for each batch process. It describes the ascending in time value for every increasing in batches number. The execution time was started from 0-51 min and the volume of batches was started from 1-500 batches.

The execution time is categorized into four zones of time which is related to the batches volume. When the batch volume is 1-5, the execution time is zero or it can be translated as a very fast time for execution or practically described as no delay in execution. This behaviour is associated with the small batches and it is the first zone. In the second zone when the batches volume was increased from 25-50, the execution time is calculated to be 1-5 min which is also described as a small time for execution. In the third zone when the batch volume increased from 125-250, the time is increased from 12-30 min and became heavy time. However, when the batches increased to be 500, the execution time is calculated to be 51 min. This behaviour can be useful

Action	Submitted Date	Job Type	Total Batches	Batches Processed	Submitted By	Completion Date
	09/01/2017 21:37	Batch Apex	500	500	hasan_awfa	09/01/2017 22:28
	09/01/2017 21:00	Batch Apex	250	250	hasan_awfa	09/01/2017 21:30
	09/01/2017 20:46	Batch Apex	125	125	hasan_awfa	09/01/2017 20:58
	09/01/2017 20:38	Batch Apex	50	50	hasan_awfa	09/01/2017 20:43
	09/01/2017 20:22	Batch Apex	25	25	hasan_awfa	09/01/2017 20:23
	09/01/2017 20:21	Batch Apex	5	5	hasan_awfa	09/01/2017 20:21
	09/01/2017 20:20	Batch Apex	1	1	hasan_awfa	09/01/2017 20:20

Fig. 4: Apex jobs

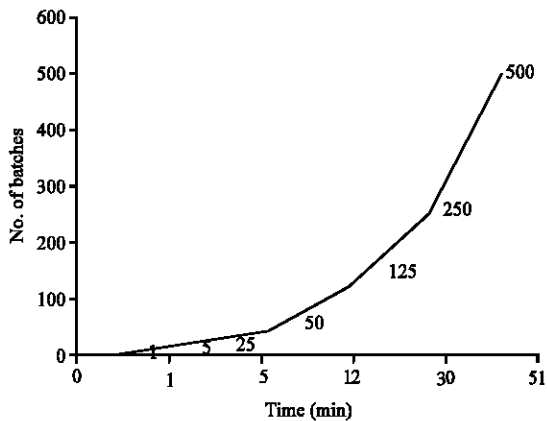


Fig. 5: Execution time consumed to implement different numbers of Apex batches

to know the consuming time for batches range (1-500) and then estimate the time if the size of batch exceeds 500.

CONCLUSION

Cloud computing is a model for enabling ubiquitous whole on-demand network access to a shared pool of configurable computing resources (e.g., networks,

servers, storage, applications and services) that can be rapidly provisioned and released with minimal management. Under the software as a service model, entire applications are made available, usually via a web interface. The customer has no influence on the platform and its underlying infrastructure.

Salesforce.com is a web-based enterprise platform offers to developers an environment for a variety of purposes. The stander MVC of Salesforce platform design pattern was used; the entire system and data schema was performed in using Objects. Visualforce was used as viewer to describe the front end to users. In Visualforce, HTML was used to show the data to users. Written in Apex language as the controller (classes) was used to represent the actions when users react with Visualforce page. Apex batch classes were used to process a large number of records and to perform different number of batches ranged between (1-500).

In this study when Apex class was developed for batch processing, the results obtained from these implementations were different in evaluating the consuming time based on the volume of batches. The execution time was increased from 0-51. The result for each invocation of a batch class placed on the Apex job queue for execution.

Managing time is an essential matter. The consuming time for part or multipart processing of batches in Salesforce's platform has a notable consideration. It's give a perceptive understanding in managing the time when need to process bunch of batches in the Salesforce's platform that exceed the permitted rolled-in batches limit.

This study recommends that the awareness of the execution time in different volumes of Apex jobs is proposed. By having this awareness the developer can schedule and maintain jobs to be executed appropriately.

REFERENCES

- Anderson, R.E., 2013. Personal selling and sales management in the new millennium. *J. Pers. Selling Sales Manage.*, 16: 17-32.
- Anonymous, 2017. Cloud infrastructure and services. MyThemeShop, Chicago. <http://pdhewaju.com.np/2016/12/17/cloud-infrastructure-and-services/>.
- Furuncu, E. and I. Sogukpinar, 2015. Scalable risk assessment method for cloud computing using game theory (CCRAM). *Comput. Stand. Interfaces*, 38: 44-50.
- Goodey, P., 2011. *Salesforce CRM: The Definitive Admin Handbook*. 3rd Edn., Packt Publishing Ltd., Birmingham, Mumbai, ISBN:978-1-78439-756-2, Pages: 439.
- Johnston, M.W., and G.W. Marshall, 2016. *Sales Force Management Leadership, Innovation, Technology*. 12th Edn., Taylor and Francis Group, New York, USA., ISBN:978-1-138-95171-6, Pages: 454.
- SD., 2007. *Soap API developer guide*. Salesforce Developers, San Francisco, California, USA. https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_quickstart_intro.htm.
- SD., 2016a. *Using batch apex*. Salesforce Developers, San Francisco, California, USA. https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_batch_interface.htm.
- SD., 2016b. *What is salesforce*. Salesforce Developers, San Francisco, California, USA. <https://www.salesforce.com/eu/crm/what-is-salesforce/>.
- Salseforce, 2016. *Apex workbook*. San Francisco, California.
- Selvakumar, C., G.J. Rathanam and M.R. Sumalatha, 2013. PDDS-Improving cloud data storage security using data partitioning technique. *Proceedings of the 2013 IEEE 3rd Conference on International Advance Computing (IACC)*, February 22-23, 2013, IEEE, Ghaziabad, India, ISBN:978-1-4673-4527-9, pp: 7-11.
- Yu, J., 2015. *Salesforce Reporting and Dashboards*. Packt Publishing Ltd., Birmingham, UK., ISBN:978-1-78439-467-7, Pages: 271.