Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 9 (2013) 381 – 387

Procedia Technology

CENTERIS 2013 - Conference on ENTERprise Information Systems / PRojMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

Agile ETL

Cristiano Xavier[a], Fernando Moreira[a,*]

[a]*Universidade Portucalense, Rua Dr. Bernardino de Almeida,541, 4200 Porto, Portugal*

Abstract

Agile ETL is a tool for technicians working in the area of business intelligence, which facilitates consolidation of information in a central repository called data warehouse. Mechanisms have been established to create, control and monitoring processes of extraction transformation and loading of data, which is supposed to give a faster response either in creating or monitoring processes. ETL Framework can quickly integrate and consolidate data with good performance indicators.

Keywords: extract; transform; load; data warehouse; business intelligence; agile; etl

1. Introduction

Companies have to respond faster to the needs because of the constant fluctuations of markets and in order to become more competitive. According to the study published by The Data Warehouse Institute, covering 510 companies, he arrival of the Business Intelligence has saved time in about 61% of companies and provided about 57% of companies with better strategic decisions, while 56% had better tactical decisions. Furthermore it has been inferred that 39% of companies generated cost savings [1].

* Fernando Moreira. Tel.: +351225572372; fax: +225570280.
E-mail: fmoreira@upt.pt

doi:10.1016/j.protcy.2013.12.042

Creating a Data Warehouse is a complex process not only in the construction of the data model, but especially in the cataloging process that goes through three phases: Extraction, Transformation and Loading [4].

ETL (Extract Transform and Load) system is much more than a tool for getting data from a source system to a central repository. It removes errors and corrects missing data; metrics provides confidence in data; data sets from multiple sources to be analyzed together; data is structured to be used by end users tools.

ETL is more than just data flow. It has the power to correct data errors and transform raw data into information that can be readily consumed by business users. In The Data Warehouse ETL Toolkit, Ralph Kimball and Joe Caserta state that the ETL portion consumes upwards of 70% of all resources required to build a data warehouse [4]

Factors like Quality of Data, Complexity of the Source, Dependencies in the Data, Logging, In-House Expertise, Support, Disk Space and Scheduling, influence the approach to loading the data warehouse, which also affects the cost of the solution [11].

For minimizing Data Warehouse implementation time, resources and cost, the proposed solution aims building a tool that automates and monitors processes of extraction, transformation and loading data. It is a framework that does not require large specialized expertise in Business Intelligence. The Framework allows creating “Out of the Box” extraction processes that put the information in a temporary area, component data transformation and loading of delete insert methodology or merge by primary key.

Building a framework that provides up functions with all creation mechanisms for extracting and integrating a Data Warehouse, is important; however, if we wish to reach a higher level, we can create new ETL processes by providing an interface layer only through a wizard,.

The main objective of the study was to get a solution to streamline the process of loading data into a central repository, Data Warehouse. For such, a prototype was created, implemented with several distinct components, such as SQL tables, a Windows service, in PowerShell functions for process automation, web interface to respond more nimbly to change orders, monitoring of extractions or integrations and data packets SSIS.

There are two ways to operate Agile ETL, using PowerShell functions or through the Website in ASP.NET.

With Agile ETL, solutions paradigm like "Business Intelligence" tends to become "Data Intelligence” [8]. This change happens because solutions become more affordable, the ability to acquire tools of analysis will not be targeted just for big companies. Besides, the way they tend to get the final consumer, even for private purposes, to analyze patterns and to draw on their data to infer and make choices more assertive [6].

This document is organized as follows. In section 2 we present related work, and in section 3 we present the prototype architecture. In section 4 we present and discuss a performance analysis of the ETL processes, and finally, in section 5? the general conclusions and the indication of future work are presented.

2. Related work

Currently in the market there are several ETL tools, which offer a range of unique features based on well-defined objectives. DataStage for example, tries to position itself through an ETL tool for high performance and availability, targeted for major companies due to cost and response capabilities with a large volume of data. SQL Server Integration Services Microsoft tries to position itself through developers ensuring a more friendly development framework [12], providing a graphical interface to build each element. Oracle Data Integrator is also another tool for ETL-oriented systems that seeks high performance and profitability and like DataStage has the handicap of being expensive tools that require great expertise for their development.

Agile ETL attempts to position itself as a low cost tool, either in the development or maintenance processes, due to its characteristics of all the monitoring, performance indicators, data aggregation and

graphical interfaces; it is possible to develop new data flows without great technical knowledge, although with great performance capabilities.

3. Architecture solution

Through automation functions in the framework, PowerShell scripts can respond by creating agile components for the entire flow, with extractors that work as an interface with the data sources to integrate data already processed ,in accordance with the business model.

The architecture is based on components that can execute Store Procedures or SQL Server Integration Services packages that put information from multiple repositories into tables in a temporary area, known as Staging Area; this data is stored in raw, to minimize the impact in operation systems [7]. After the data is temporarily stored in Staging Area, it is time to work the information in order to consolidate according with the business rules into Data Warehouse. Finally, the data processing is performed on a multidimensional repository to provide an access method, visualization, and analysis with high performance and flexibility.

3.1. Technologies

The architecture is based on Microsoft tools: Microsoft Integration Services, Microsoft Windows Service Applications [5], Microsoft SQL Server, Windows PowerShell [3], ASP.NET and Microsoft SQL Server Analysis Services [10]. These tools were chosen because of licensing issues and greater technical knowledge in this type of technology.

3.2. Agile ETL essential characteristic

Agile ETL framework created to perform the extraction, transformation and integration of data in Data Warehouse, through orchestration process that includes a model of indicators, processes for data aggregation, scheduling ETL processes and OLAP processing.

3.2.1. Extractor

The extractors are the objects responsible for loading data from data sources, OLTP, XML files, Excel files, etc., to a Staging Area; it is the responsibility of each extractor to know where and how to put the information in the tables of Staging. At this stage, there will be typically data transformation; performance issues with connectivity to the operation systems should be minimized as much as possible to avoid negative impacts that may exist in the systems.

Currently there are two types of extractors: SSIS package or SQL store procedures. The framework provides a generic component SSIS that, by changing the variable values and the connectivity of the source, may be reusable in more than one packet extraction.

3.2.2. Integrator

Integrators have the task of loading data from Staging area to Data Warehouse and of transforming the data in a consolidated manner to respond to business model.

The integrations can be of two types, SSIS packages or SQL procedures. The framework takes more advantage of SQL procedures (there are major limitations in SSIS packages in code reuse), thus, they were constructed generic procedures through dynamic SQL, designed in two distinct types of integrations. Strategy deletes records before inserting, is typically used for metrics that make a massive insertion, erases records

included in their partial extraction earlier, or if the extraction is complete, erases the entire table and inserts new records.

There is another strategy for loading data that never deletes records in data warehouse, and only inserts new records or changes data who have been registered before; this load requires a verification key that controls through the primary key if the record is new or not, and if it is used mainly in the analysis dimensions or metrics that have changes throughout its existence.

3.2.3. Indicator Model

The model is an agile architecture indicators storage metrics that enables the system to incorporate new indicators without having to create new tables or new groups of metrics in OLAP. With distinct structures and resources in multiple data sources, the consolidation goes through all the information collected and centralized in a table, with a generic scheme, where all the information will serve a single group of metrics in OLAP.

3.2.4. Scope

Scope is a feature in Agile ETL, to restrict the scope of extractions, and segmental information not only by partial extractions, which also reduce the scope for extraction, but also by the start date and end date of registration. The concept of Scope exists in framework to respond in a responsive manner to small changes in the extraction processes. An example of a case study is an extraction of alarms on a server where there are recurrent events of warning type, since the excessive information can cause disinterest; when demand becomes more difficult, it is helpful to filter temporarily the alarms drawing, and ideally only extracting critical alarms.

With Scope it is easy to add items' restriction in SQL queries without having to change the SSIS package and still disabling when no code changes.

3.2.5. Aggregation data feature

As storage in enterprise systems is expensive and analysis of historical data often do not have great need of detail; the model of indicators has native functionality of indicators aggregation , such as providing more disk space in the servers where Data Warehouse is housed.

The timing has two distinct options of aggregation. Aggregation at the hour adds records regardless the level of detail at the time, in other words, in the analysis of a performance indicator "Occupancy Rate of CPU", the agent system is collecting information from minute to minute and a server specifically for ETL process saves this information in data warehouse, with this detail. With aggregation process and specific timing model indicators in selected period data is replaced with a level of detail to another baseline, by aggregating their averages, sums and number of records inserted at the time. With this aggregation it has lost the down level detail, with decreasing number of records in the Data Warehouse. With aggregation, the maximum level of detail of information may be irrelevant in most of the cases, because knowing the "CPU Load Factor" in a particular minute to five years ago does not bring a significant added value compared to the cost of storage.

Currently there are two types of aggregation, bundling the hour and day time.

3.2.6. Logging

There is a centralized event table to add all the information and status of the Agile ETL framework. Log table should contain information on all processes like monitoring messages, warnings or errors in the application.

3.2.7. Web Site

Never losing the context of the framework, and always targeting the application for a specialized audience, as a tool to aid in the task of building and maintaining centralized repositories often called by Data Warehouse, it is available through a web interface that invocates PowerShell scripts to create the new ETL processes [9]. In addition to creating new SSIS packages, new tables, views and procedures in SQL also allows maintenance and configuration of extraction processes and integration. The timing, monitoring and process automation with portability, goals were achieved with this interface.

4. Performance

Knowing that the agility of construction or the maintainability of ETL processes have stronger visibility, it is necessary to demonstrate the gains also in execution performance. For that, it was held a performance analysis, for the integration and extraction tasks. We carried out a survey of a sample period between 2012/11/01 and 2012/12/01, and made up twenty-nine respective extractors and integrators, each one with its timing.

The results were analyzed in comparison to other ETL technologies, based on a benchmark study [2]. ETL tools used in the performance comparison are TOS – Talend Open Integration Solution, PDI – Pentaho Data Integration, DataStage – IBM InfoSphere DataStage and Informatica – Informatica PowerCenter.

4.1. Extraction

The extraction processes analyzed are all made by Agile ETL using SSIS packages. These packages are aimed at the transition of records from their origins to the staging area.

There are several distinct sources in the analysis, many servers that use the database in MS SQL Server 2005, MS SQL Server 2008, MS SQL Server 2008 R2, Oracle 10 and Oracle 11.

As can be seen (Table 1), with the ETL framework, the extraction of data through SSIS packages had quite satisfactory results, placing second extractions in excess of one hundred thousand records and third extractions in ten thousand records.

Table 1 - Benchmark Extraction Data

Framework	10,000 (rows)	100,000 (rows)	5,000,000 (rows)
TOS	1 (s)	7.8 (s)	39.1 (s)
PDI	2 (s)	15.5 (s)	83.8 (s)
DataStage	3.4 (s)	12 (s)	40 (s)
Informatica	40.67 (s)	45 (s)	54.33 (s)
Framework ETL	3.1 (s)	11.2 (s)	35.3 (s)

4.2. Integration

The integrations analyzed are all made by ETL framework, using procedures in SQL. These procedures aim to consolidate the information in the data warehouse, making changes to data stored in Staging area, aimed at responding to the schema designed and engineered in the Data Warehouse. Integration processes haves made an adjustment performance by creating indexes, partitioning tables, ensuring better results in the transformation of data.

In Table 2, we can verify that framework ETL, with data integration through SQL procedures, had very positive results, always getting in the top three in all tests.

Table 2 - Benchmark Integration Data

Framework	10,000 (rows)	100,000 (rows)	5,000,000 (rows)
TOS	56.51 (s)	68.1 (s)	199.26 (s)
PDI	369 (s)	407 (s)	496 (s)
DataStage	24 (s)	30 (s)	55 (s)
Informatica	96.5 (s)	109 (s)	232.5 (s)
Framework ETL	40.69 (s)	70.81 (s)	167.2 (s)

5. Conclusion

The prototype implemented centralizes the creation and automation of ETL processes, enabling the construction of central repositories of data to companies with less financial capacity.

In the implemented prototype it was possible that some adjustments further sensing facilitate migration information processes. The idea would be not only making the migration table to table almost from the origin to the Data Warehouse, but also designing a new component like the "Data Source View" SQL Server Analyses Services, which defines all the desired structure through "drag and drop" to import various objects, from different sources; the model would be automatically adjusted in response to schema designed for the Data Warehouse.

Another future goal is to integrate the framework with a mobile interface that allows monitoring and creating new components via a smartphone, tablet be it IOS, Android or Windows Phone.

In the long term there is the intention to add a new architecture framework for ETL, an event-driven architecture using a subscription system event, which unlike the current system has got an architectural request and response between servers, (even in case they are original ones), Staging and Data Warehouse. The event-driven architecture works through subscription of events, allowing subscription and performing actions in response to events created. One of the features valid for the framework would be removing schedule task in the extraction processes; making only subscriptions on the operation systems servers. Entries of new records or changes to the data, would be automatically routed to the Staging area and then processed and entered into the Data Warehouse.

References

[1] Eckerson, W., 2010. Smart Companies in the 21st Century. Seattle: The Data Warehousing Institute.
[2] Infosphere. 2011. ETL Benchmark Favours Datastage and Talend. Obtido de ETL Benchmark: http://it.com/blogs/infosphere/etl-benchmark-favours-datastage-and-talend-28695
[3] Holmes, L., 2010. Windows PowerShell Cookbook. Sebastopol: O'Reilly Media.
[4] Kimball, R., Caserta, J., 2004. The Data Warehouse ETL Toolkit. Indianapolis: Wiley Publishing, Inc.
[5] Microsoft., 2010. Introduction to Windows Service Applications. Obtido de MSDN Microsoft: http://msdn.microsoft.com/en-us/library/d56de412(v=vs.80).aspx
[6] Minelli, M., Chambers, M., Dhiraj, A., 2012. Big Data, Big Analytics. New Jersey: John Wiley & Sons, Inc.
[7] Ndlovu, S. W., 2011. Programmatically Create Data Flow Task in SSIS Package Using C#. Obtido de select Sifiso: http://www.selectsifiso.net/?p=288
[8] T. Moss, L., Atre, S., 2012. Business Intelligence Roadmap. Boston: Addison-Wesley.
[9] Sojo, E., 2012. Creando paquetes de SSIS con .NET. Obtido de Blog de Eduardo Sojo: http://eduardosojo.com/2012/01/03/creando-paquetes-ssis-con-net-creando-data-flow-task-y-elementos-internos/
[10] Thomsen, E., Spofford, G., Chase, D., 1999. Microsoft OLAP Solutions. Indianapolis: Wiley Publishing, Inc.
[11] Turban, E., Sharda, R., Delen, D., King, D., 2010. Business Intelligence (2nd Edition). London: Prentice Hall.

[12] Webb, C., Ferrari, A., Russo, M., 2009. Expert Cube Development with Microsoft SQL Server 2008 Analysis Services. Birmingham: Packt Publishing Ltd.