# ELTMaestro for RedShift: ELT in the Cloud

## Author's note

*More often than not, I find it hard to figure out what tech companies or their products do by looking at their web pages. I end up trying to deduce what they do by looking at their Wikipedia pages, reading unfriendly reviews of their products, finding out who they compete with, or trying some other way of backing into an understanding. It almost never turns out that what they do is inherently difficult to understand. Rather, it's a combination of the following:*

- *I lack context. I'm not familiar with the problem they're trying to solve.*
- *They are desperate to demonstrate that they are familiar with all of the buzzwords, and in the process, they lose me (and, I suspect, 95% of their audience).*
- *The scope of the problem they claim to solve is so large that I have no idea what they actually do: "Finally! A software package that puts the right information in front of the right person, right when they need it!" In other words, they've solved all problems of computer science.*

*I've begun my white paper this way because I want to explain a new product to you, and I want to avoid these mistakes. In fact, I promise to avoid these mistakes! I will tell you up front what problem we are trying to solve. I'll keep my buzzwords to a minimum, and define them carefully. (As far as I know, all my buzzwords are pretty passé.) And while I believe that the product that I'm telling you about is useful and interesting, and solves some important problems in clever ways, I will refrain from claiming that it solves all computer science problems.*

*Does our product solve your problem? That's hard to know at this point. But you might want to keep reading, because in the course of explaining our product to you, I'll also explain some things about data warehousing, business intelligence and cloud computing. So if those things are interesting, stay tuned!*

## Introduction

Data warehousing in the Cloud is a new technology that promises to bring significant improvements in cost and efficiency to the business intelligence enterprise. The new technology also introduces disruptions into established IT practices, because high-level tools and languages for building data warehouses are no longer available. This lack makes it difficult to take advantage of the advancements of the new technology. In this paper we review data warehousing technologies and architectures, and discuss in detail why the best architectures are disruptive. We then introduce ELTMaestro for RedShift as a solution which allows practitioners to take full advantage of the savings and efficiency of the new technology without sacrificing the convenience of high-level data warehouse building tools.

## A Buzzword Glossary

Cloud: In the old days (which, of course, are still the way most of the world works), computing services were provided by *data centers.* What are computing services? Databases are a typical example. A company might decide that it needs a database with certain specifications, and the data center would then arrange for that database to be provided – that is, the data center would set up the database server, install and configure the software, load the data, and maintain the database. Employees of the

company who were *users* of the database would then communicate with the database through a non-administrative interface; administrative matters such as backups, upgrades, and so on would be carried out by data center personnel, and hopefully be transparent to the user. The data center would generally be organizationally part of the company, or under the company's control.

The *Cloud* refers to a new way of organizing and providing computing services. In the Cloud paradigm, the computing resources are owned and operated by a separate entity. The computing resource resides somewhere on the web. Unlike data centers, Cloud service providers don't contract to provide access to certain pieces of hardware running a certain pieces of software; rather, they undertake to provide a services (such as a database with certain characteristics) for a certain period of time. Cloud services are generally provided on a subscription basis. Other examples of Cloud computing services include web servers, file servers, and other specialized kinds of computing services such as speech recognition.

Data warehouse: Companies use databases for their daily operations. A bank, for example, will have a database of bank accounts. A row in that database represents the account of a particular customer. When the customer deposits or withdraws money from the account, that row gets updated. Similarly, a retail store chain will update a database whenever a customer makes a purchase. Records are maintained of every item sold at every cash register at every store. Databases like these, that are used for the daily operations of a company, are called *operational* databases.

These same companies find that they want to *study* and *analyze* the information gathered by their operational databases. The bank might want to know the average balance of customers in a certain zip code; the retail store might want to know how diet soda is selling during a certain holiday. It generally is not possible to get this information directly from the operational databases. The operational databases are critical to the operation of the company, and the cost of interrupting them is very high. The kinds of database queries necessary to gather analytical information are fundamentally different from the kinds of queries those databases usually handle, and running those queries risks compromising the operation of the database. Furthermore, the operational databases don't typically have all of the information necessary to answer the analysts' questions.

For these reasons, information is typically gathered from operational systems and copied into a different kind of database, one which is optimized to support study and analysis, and answer questions like "how much diet soda is selling during a certain period," or "what is the average balance of customers in a certain zip code." Such a database is called a *data warehouse.*

Business intelligence: Business intelligence sounds as though it means trying to find out what other companies are doing. In fact, it usually means trying to find out what your own company is doing. Business intelligence is information such as "how much diet soda we sold during Labor Day weekend."

Business intelligence (BI) and data warehousing are often used more or less interchangeably. An accurate statement would be: A desire for business intelligence is the reason most data warehouses are built.

ETL: ETL (Extract – Transform – Load) is the usual process by which data enters the data warehouse. Data is *extracted* from operational databases and other sources, *transformed* and combined with other data into a form suitable for the data warehouse, and then *loaded* into the data warehouse.

ETL is, in principle, an abstract process with no implication as to how it is implemented. In practice, however, ETL is usually taken to imply the following architecture:

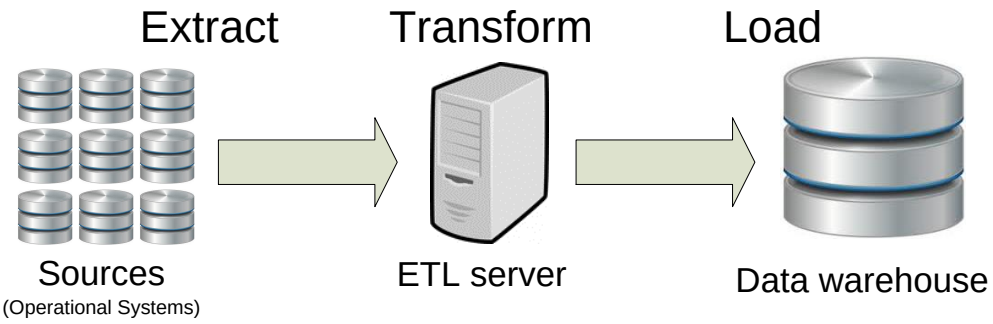


Figure 1.

Typically, an ETL system runs on a powerful *ETL server* (or group of servers). An ETL software program will coordinate all of the component ETL processes: extraction from sources, transformation on the ETL server, and loading into the warehouse. Most ETL processes run periodically, typically once a day, and in the dead of night, when there is less demand on the operational systems, since even copying data from the operational systems to another database may cause interruptions. Some ETL processes run continuously, throughout the day.

The growth in the size of data warehouses has been nothing short of explosive in recent years, and the demands on ETL systems to handle large data volumes have grown commensurately. In general, ETL vendors have answered this demand by taking advantage of opportunities for parallelism. Generally, the parallel processes in question must run on the ETL server. As a result, the ETL server – hardware together with software – has become a critical and expensive component of many companies' IT infrastructure.

ELT: ELT (Extract – Load – Transform) is an alternative to ETL. At a simple level, it just means that we reverse the second two steps – we load the data into the platform containing the data warehouse, and *then* we perform whatever transformations and re-combinations of the data are necessary to get it into the form necessary for the data warehouse. To understand why this is an interesting idea, however, we need to make a brief digression into another aspect of data warehouse technology development.

Loading data warehouses isn't the only challenge that has resulted from the growth of data warehouses. Their growth has also made it increasingly difficult to get useful answers from them. When people first started implementing data warehouses, they implemented them on the databases that were available, which happened to be databases tuned for *on-line transactional processing,* or OLTP. This is the kind of processing that operational databases do, and that's what most databases were good for. An OLTP-tuned database will be good at handling lots of "small" transactions – operations that involve only a few records at a time. This corresponds to the scenario of many bank customers updating their individual accounts, or many retail customers making individual purchases.

The use of a data warehouse, by contrast, is much different. The data warehouse is expected to handle only a few queries (because the number of analysts working for the company is much smaller than the

number of customers). But their queries are huge – they may involve looking at every customer record the firm has.

It turns out that databases can be tuned to handle OLTP-type queries well, or they can be tuned to handle data warehousing queries well, but not both. This became evident to companies that had built their data warehouses on traditional OLTP databases – their databases were not well suited for data warehouse-style queries. When they started to pose more complex questions to their data warehouses – questions that required bigger joins, and involved a more significant fraction of the records in the database – the queries stopped returning results.

To solve this problem, data warehouse architects began to replace the databases that hosted the data warehouse with a new kind of database – one that was tuned to handle the large queries associated with data warehousing. In many cases, this new database was a *data warehouse appliance* – a special piece of hardware that served as an ultra-efficient data warehouse. Netezza (now IBM PureData for Analytics) was an especially successful example of this type.

An interesting side effect of the switch to powerful data warehouse appliances was that architects began to question the utility of ETL servers. The things that ETL servers did well, such as massive sorts and joins, parallel processing, and efficient loading, could all be done even better by the new special-purpose data warehouse platforms. This begged the question – why not just load the data directly from the source systems to the target system, and then complete all necessary processing there – extract, load, and transform?

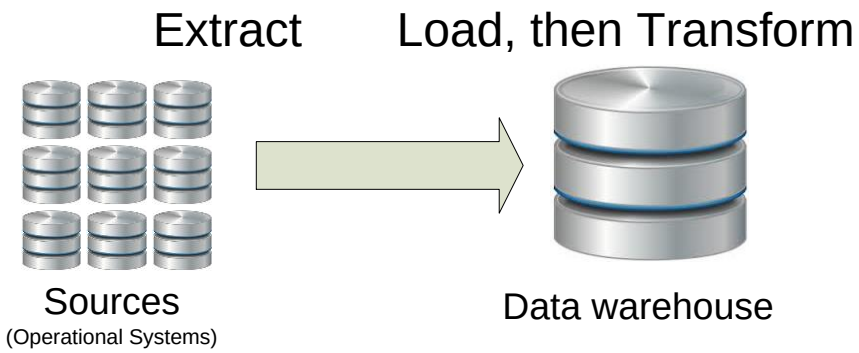


Figure 2.

## Why Not ELT in the Cloud?

Even a casual comparison of Figure 2 with Figure 1 tells you that Figure 2 is cheaper, faster, and easier to maintain. In fact, it's difficult to overstate the delta between the two. To take one example, it isn't uncommon for ETL vendors to price their systems according to the number of processors *in the target.* Special-purpose data warehousing platforms are generally MPP (massively parallel processing) systems. Larger models contain upwards of a hundred processors. That means that the ETL server and software can cost tens of millions of dollars. Figure 1 can cost a multiple of Figure 2 – tens of millions of dollars more.

The Figure 2 architecture will also be faster, for two reasons: First, there is one less hop between machines. Such hops are inevitably the worst bottlenecks of any ETL system. Second, however

powerful a transformation engine the ETL server and software may be, it is likely that they are outperformed by the data warehousing platform.

The reasons in favor of ELT, then, when the target is a special purpose data warehousing platform, are very compelling. Why is ELT not the norm? In the rest of this paper, we'll consider the case where the target is Amazon RedShift. RedShift is a powerful Cloud-based MPP data warehousing platform. It is similar in many respects to data warehousing appliances like Netezza, except that it resides in the Cloud.[1]

## Impediments to ELT

If Figure 1 costs millions more than Figure 2, there must be *some* reason why the world hasn't gone from Figure 1 to Figure 2. In fact, there are several.

### Inertia

Most ETL systems are not new. We may be able to look at Figure 1 and see that it represents a vast waste of money and resources over Figure 2 – but chances are, Figure 1 didn't spring into life in its current form. Perhaps when the ETL server was originally purchased, the target was different, and ELT was not an attractive option at that point. In fact, many companies changed their data warehouse platforms from traditional databases to special purpose data warehouse platforms without changing their ETL systems. Their ETL systems may have been expensive, but in many cases, that money was already spent (or at least, so it appeared.)

### Change our ETL systems…to what?

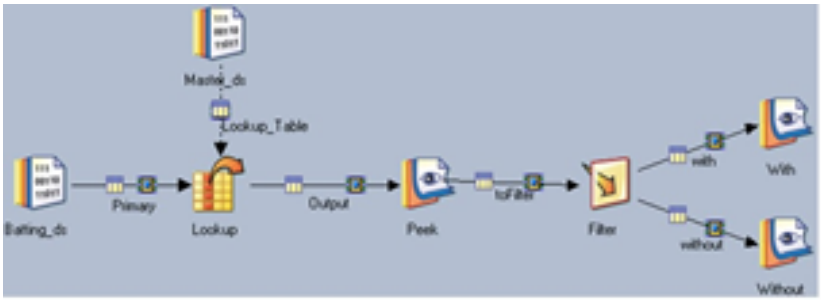


Figure 3.

Another reason companies don't want to change their ETL systems is that an entire ecosystem of applications, best practices, specialties, training programs, consultants, and terminology have grown up around their existence.

The diagram in Figure 3, for example, was produced by IBM's DataStage, one of the most ubiquitous ETL programs. Such a diagram is called a *dataflow* diagram; we say that it is constructed in a *visual dataflow language.* Most ETL practitioners are used to working with diagrams of this type. Visual dataflow languages are far from the only thing that ETL programs offer users to make their lives easier. ETL programs also come with connectors all of the sources where people keep data that they might want to

[1] In a previous white paper, ELTMaestro and the ELT Paradigm Shift we considered the case where the target was Netezza.

put in a data warehouse. And perhaps most importantly, ETL programs allow programmers (and enterprise schedulers) to view ETL operations in terms of *jobs* and *batches* – in other words, periodic invocations of sets of programs on successive datasets. Reliable job and batch abstraction is necessary for a production system.

All of this exists in the Figure 1 world. In the Figure 2 world, it would all need to be constructed from scratch, with none of the support (best practices, consultants, training programs, etc.) above. To the ETL programmers, going from Figure 1 to Figure 2 would be like going back in time – all of the advanced tools that make their lives easier would disappear, and they would have to program in languages like SQL, the way people did ETL 15 or 20 years ago.

### Moving data into the Cloud

When the target data warehousing platform is RedShift there is an additional problem that needs to be considered. This is not a problem of Figure 1 vs. Figure 2, but rather a general problem of large scale data integration in the Cloud.

In the pre-Cloud world, we would generally assume that all of the components of Figure 1 or Figure 2 are implemented in a single data center. The sources would be connected to the target by an internal high-speed network. All of the machines in question are behind a firewall.

When Figure 2 is implemented in the Cloud, these assumptions do not hold.

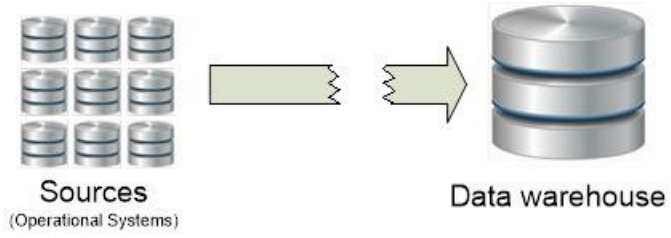


Figure 4.

The network connecting the sources and the data warehouse is the internet. This raises several concerns:

- Communication between the sources and the data warehouse are vulnerable to eavesdropping.
- Illegitimate actors may attempt to connect to the source or target.
- Bandwidth is no longer under the control of the enterprise.

## ELTMaestro for RedShift

ELTMaestro for RedShift is a software product available by subscription on Amazon Web Services Marketplace. ELTMaestro for RedShift addresses the problems raised in the sections above in order to make ELT a truly workable, cost-effective solution for the Cloud.

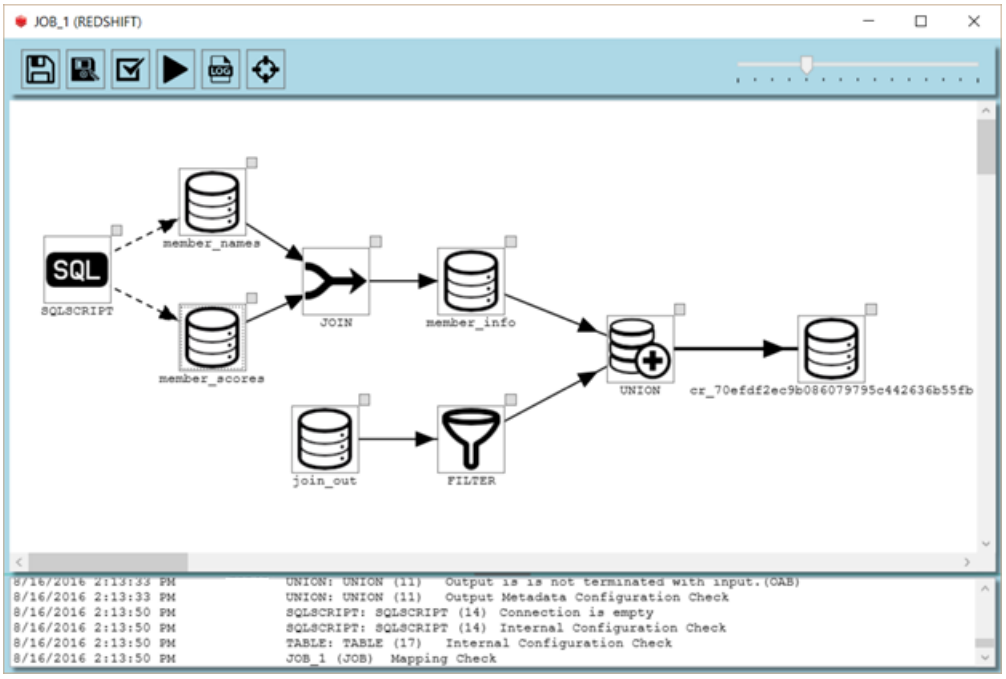


Figure 5. ELTMaestro for RedShift screenshot.

ELTMaestro allows data warehouse builders to leverage the cost savings and efficiency of the Figure 2 architecture without giving up the advantages of the ETL ecosystem. ELTMaestro for RedShift:

- Provides users with an intuitive visual dataflow language. Users familiar with other ETL interfaces are able to use it without training.
- Presents users with a job/batch abstraction. ELTMaestro fully supports enterprise production operations.
- Supports connection to most widely used source systems.
  Compiles dataflows into efficient RedShift SQL, and takes full advantage of RedShift's speed and capacity.
- Has strong data quality analysis and control capabilities. The designers of ELTMaestro are long-time practitioners of ELT and data warehousing and have leveraged their experience to produce a product that includes features they felt were missing from other systems.

## A lightweight solution, powered by RedShift

ELTMaestro for RedShift provides users with power and feature capability comparable to ELT systems that are orders of magnitude more complex and costly. ELTMaestro is actually a low-cost, lightweight tool which installs in minutes. How is this possible?

Traditional high-performance ETL systems are principally composed of a *user interface*, a *repository* where jobs, batches and other artifacts are stored, an *engine,* which does the actual work of parallel operations on data, and a *compiler,* which translates user instructions (such as the visual dataflow language) into instructions that the engine can execute. Of these the engine is by far the largest and most complex component. The engine is generally responsible for distributing operations like sorts,

filters, and joins among multiple processors, and making sure that they proceed as efficiently as possible.

ELTMaestro for RedShift is lightweight because it uses RedShift as its engine. Instead of devoting design effort to looking for the fastest ways to do sorts and joins, ELTMaestro's designers look for the best ways to leverage RedShift's excellent capabilities for high volume data operations. The result is that the speed and performance of ELTMaestro for RedShift is essentially the speed and performance of RedShift.

### Agents control data movement

ELTMaestro addresses the problem of moving data into and across the Cloud by using a scheme of *agents.* Agents are processes installed on data sources that know how to communicate with the target. Any number of agents may be installed on a source, and any number of agents can be installed overall. When a connection to a source is requested, the first responding agent that can support that connection is used. Multiplexing may occur when another connection to the same source is sought; in this case another, unused agent may reply and support the second connection. By supporting multiplexing, agents help the enterprise maintain a degree of control over bandwidth in a Cloud implementation.

Agents take care of encrypting and compressing data communication between source and target, alleviating the concern of eavesdropping.

Agents also help to maintain data security because the data sources where they reside tend to be inside internal networks where they are invisible to the outside world. The agent can see the target machine, and can initiate communication with the target machine, but the target machine can't see the agent, other than to respond to its queries. Agents maintain contact with the target machine by pinging the target every few seconds, to see if there are requests for their data. If there are, a connection is established and data is sent to the target. The target never needs to know the IP address of the source, which helps to prevent illegitimate connections.

## Conclusion

Use of an advanced Cloud-based data warehousing platform such as RedShift opens the possibility of an ELT architecture – what we've referred to in this paper as "Figure 2." The ELT architecture with RedShift presents very compelling cost and performance advantages. It is difficult to implement in practice, however, because of the lack of environmental infrastructure – the tools that people are used to working with are lacking. As a result, an ELT implementation would require a significant code base to be developed from scratch, and the associated risks tend to outweigh the ELT-associated cost savings.

ELTMaestro for Redshift addresses this lack by providing the required tools in the ELT context. With ELTMaestro, the necessary infrastructure does not have to be built from scratch; jobs can be developed, tested, and deployed as quickly as they are with traditional ETL systems, and at considerably less cost. The considerable advantages of ELT can now be realized without paying a price in terms of tool quality.