

Evaluation of two ETL's¹ : CloverETL vs. Talend Open Studio

To follow our previous article about ETL's introduction, we will present and compare two open source ETLs: CloverETL and TOS (Talend Open Studio).

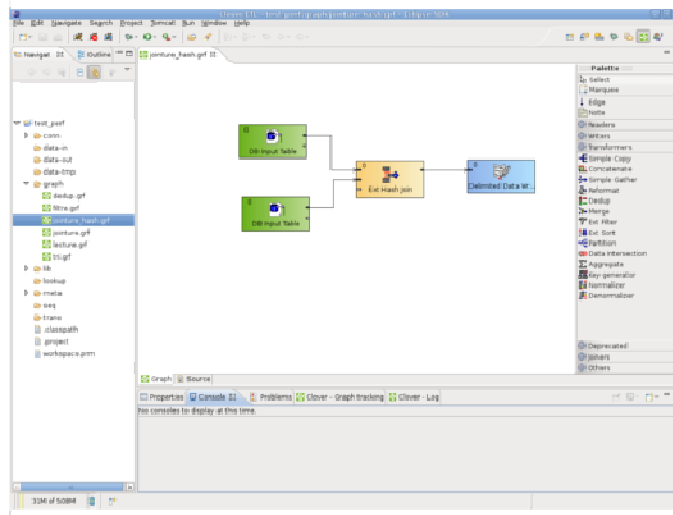
Original:

<http://www.axege.com/Evaluation-de-deux-ETL-Clover-ETL-vs-Talend-Open-Studio.html>

¹ Extract, Transform, Load

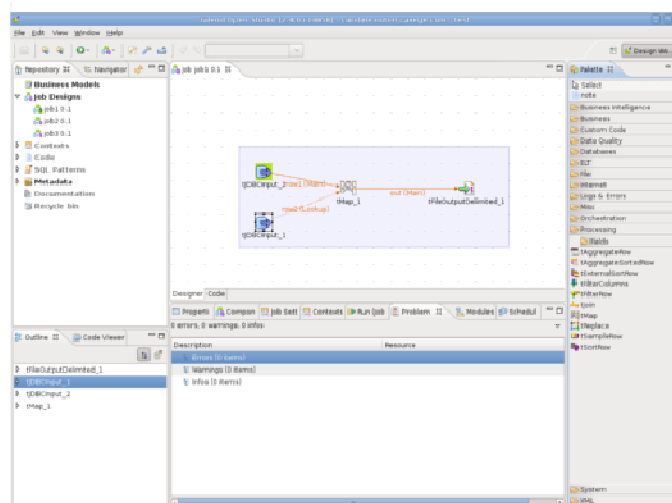
Presentation

Clover consists of two parts: CloverETL, which is an engine, and Clover.GUI², graphical interface facilitating the creation of data flows. Both are based on Java technology. They are platform independent and resource efficient. Clover.GUI is available under commercial license and is supplied in the form of an Eclipse Plugin. On the other hand, Clover.ETL engine is provided under LGPL³ license and can co-operate with any tools (even with commercial licenses).



Clover.GUI

TOS is a code generator. Its interface allows transforming data flows into graphical representations, which automatically transcribe to Perl or Java code. They can also be exported and run without TOS. This ETL is distributed as an installation package. This tool is provided under license GPL and it can not be embedded into any software without this license.



² Graphical User Interface
³ Lesser General Public License

Evaluation

For the needs of their clients, Axège has elaborated a comparative study of Clover.ETL and TOS. The first one is widely used by developers of Axège Santé, the second is very common in the market.

Methodology

The process of evaluation conforms Business Readiness Rating. Four phases of the method BRR are following:

- To do a quick evaluation to create a short list of software to be valuated
- To identify categories and metrics of the evaluation
- To collect the relevant data
- To rate these data from 1 (low) to 5 (high)

Study

The study has been realized from the technical point of view. Its benefit is to analyze the software from various angles rather than in a general overview.

It was realized with the following configuration:

- AMD⁴ Athlon(tm) 64 x2 Dual Core Processor 4400+
- 2 GB RAM
- Ubuntu version 8.04 (Hardy)
- TOS version 2.3.3
- Clove.GUI version 1.9.2 and Clover.ETL 2.4.3

The categories with descending importance:

- Functionality: coverage of software functionality (metadata, transformations....)
- Performance: memory consumption and execution time
- Documentation: quality of the software documentation
- Usage spread: usage of the software in the market of ETL

⁴ Advanced Micro Devices

- Professionalism: applied methods in the process of the development and of the organization of the project
- User-friendliness: quality of user interface
- Community: level of activity of the user / developer community
- Architecture: modularity, portability, flexibility, scalability, ease of integration
- Packaging: number of supported platform
- Maturity: age, stability, history and fork
- Quality: quality of the conception, the code and the tests
- Services: support and service

Comparison and results

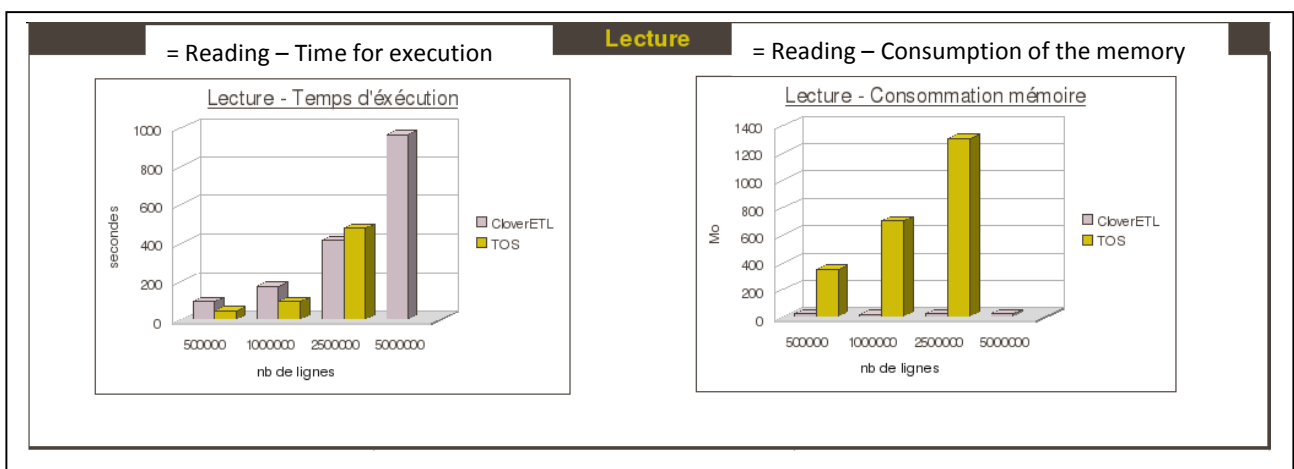
Study category by category:

Functionality:

TOS has larger palette of components (246 components against 57), so it provides more functionality. This fact doesn't discriminate Clover.ETL in its role of ETL tool, it has some components that TOS is missing. For example, Clover.ETL has the component „DataIntersection“. It allows the intersection of two flows - A and B - based on the specific key to be done. Three outputs are present on this component: the records present only in A, the records in A and B, and the records only in B.

Performance:

Clover.ETL has an advantage because TOS consumes a lot of memory. Despite TOS' good execution time on a small number of processed records (up to 2 million) its huge memory consumption doesn't allow it to read more than 3 million records. Here are some of the results:



Documentation:

The documentation of TOS is incomplete. Lots of components aren't described in the user manual and some explanations aren't so precise.

Usage Spread :

From its beginning TOS has been downloaded 250 000 times, but Talend estimates that it actually has about 75 000 users. TOS hugely invests into marketing, so it is more widespread than Clover.ETL in the world of ETL.

Professionalism:

TOS is more organized than Clover.ETL from the view of modification and extension of the code. For example, TOS uses roadmaps, so new functionality and versions are better managed.

User-friendliness:

TOS has a very pleasant and comfortable interface because many actions can be done by drag-and-drop. Clover.ETL is easier to use because of fewer components and is therefore clearer.

Community:

We can see from the activity on forums that TOS has a very active and participating community.

Architecture:

The code of Clover.ETL is more understandable. So it is easier to be modified.

Packaging:

Both ETL can be used on more systems like Windows, Linux, Debian, Unix Solaris, Mac OS X...

Maturity:

Clover.ETL is a little bit older than TOS but both don't come from fork and have very little possibility to fail.

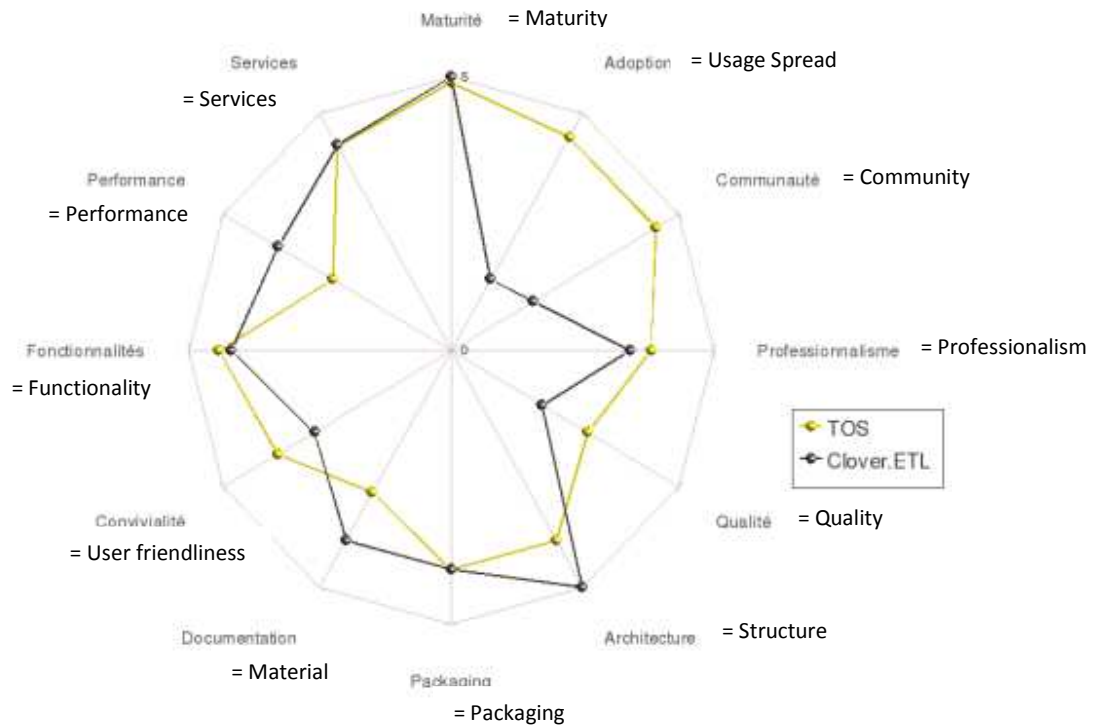
Quality:

Both have the bugtrackers, but only TOS uses it.

Services:

Each organization offers the solutions of support and service. Their offers are organized on expert level and according to the size of the company.

Conclusion



None of these tools are better than the other: summary of the advantages and disadvantages of the TOS and the Clover.ETL.

	<i>TOS</i>	<i>Clover.ETL</i>	<i>Explanation</i>
Functionality	x		Bigger palette of components for TOS but Clover.ETL is a complete ETL.
Performance		x	TOS needs a larger part of the memory and doesn't finish all jobs within the limit.
Documentation		x	Material of TOS is incomplete and inaccurate.
Adoption	x		TOS has a good marketing and big community. Clover.ETL community doesn't exist.
Professionalism	x		TOS is more organized than Clover.ETL in modification and code extension. TOS has better

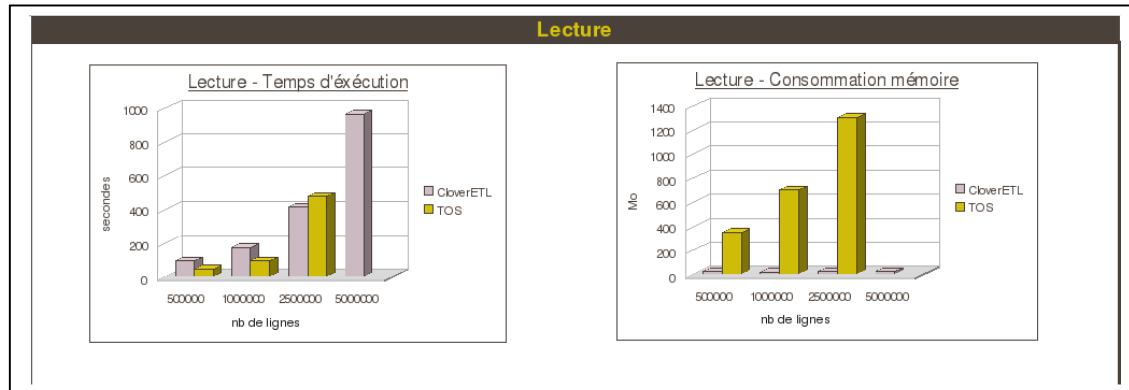
			project management.
User friendliness	x		Interface of TOS is nicer. Clover.ETL is more easy-to-use.
Community	x		TOS has active and participating community.
Architecture		x	Code of the Clover.ETL is easier to modify.
Packaging	x	x	Both can be used on many systems.
Maturity		x	TOS is more recent than Clover.ETL.
Quality	x		TOS uses a bugtracker.
Services	x	x	Both offer various level of support.
License		x	The license of TOS (GPL ⁵) is very restrictive.

None of these ETL tools is better than the other so we can conclude that the choice between them should be done according to the customer's needs.

⁵ General Public License

Observation and comments from a CloverETL consultant

Appendix A: Scalability Observation



We believe that the performance measurement conducted by Axege, depicted by the two performance graphs above, clearly shows that **TOC is not a scalable solution and cannot operate on larger data volumes**. Good scalability is probably the most important characteristic of any ETL tool as it gives a clear idea of how the tool behaves with growing volume of input data, thus showing if the tool will be capable of processing large data volumes once deployed.

Lack of TOC scalability can be easily observed on the graph „Consommation memoire“. **With growing data volume**, not only the processing time grows, the **memory consumption increases as well** with linear scale. At 2.5 million rows of input the TOC requires 1.4 GB of memory - almost 580 kB per single input line! At 5 million rows the **TOC runs out of memory and fails to process the input completely**: according to hardware specification the system has 2 GB of physical memory, while TOC would need 2.8 GB of memory according to our calculations.

(5 million lines * 580 kB per line= 2.8 GB).

We would also like to provide reader with an estimate of physical data volume processed in this comparison study. An exact calculation requires knowledge of input file format. Unfortunately this information is not part of the study, therefore we have to compromise with an estimation. Let's consider an ASCII text file with fixed-length record format, having 32 bytes per line. For an input files with 5 million rows we calculate its size: 5 million * 32 bytes = **152 MB of input data**.

From our experience **150 MB** of input is a relatively small data file and **real-world scenarios** operate with data volumes that are easily **100-1000x larger**.

An excerpt of such file could look like this:

```
ID Cust Amount Cur
1 42385 35665.20 2
2 34210 36134.06 2
3 18495 35907.54 32
4 21780 8505.68 6
```

This is why **CloverETL** is designed to run with **fixed memory consumption** and **scalability** in mind. Once again, this statement is clearly proven by Axege graphs. The graph „Consommation memoire“ shows **CloverETL consumes less than 50 MB** of memory **for any data volume**. Processing time grows linearly with data volume: if processing of 1 MB takes 1 second, processing 10 MB takes 10 seconds.