

Systems' lifelong logs approach

This approach can be used for finding an automatic solution for already clustered error. A human interaction will not be necessary in the process.

Firstly, for each insider, we will store a list consisting all actions taken starting after their operating system installation. An action could be:

0. Operating system installation [distribution, version, flavor]
 1. Package installation via apt-get.
 2. Changing a DPCS protected configuration file. [which lines was changed]
 3. Execution of a DPCS protected application [return code, parameters, log]

They will be stored as “SLL/username-computer.ss” structured streams on the HDFS and will be incrementally updated every a few days.

Daily, during the night, SLL algorithm will look at the action files and try to cluster the errors. If a specified error will have a large enough representation, the SLL algorithm will try to find a representation of two classes:

1. Actions “run the application” resulting with a specified error.
2. Actions “run the application” resulting with a success

The next step will be to discover statistically significant differences in each route from system installation to a described action between these groups. We will start with checking, what package always was installed before a successful action and what package was always missing before a failed action.

If the difference could be reduced to a single package, we have a solution: An user have to install this package in order to make this command work without an error.

Example

Let's look at the SLL algorithm on a simple example from my personal work.

I've tried to convert an ipython notebook to a PDF format, using a command:

```
nbconvert --format=pdf lecture.ipynb
```

It have failed with an error:

```
[NbConvertApp] CRITICAL | Bad config encountered during
initialization:
[NbConvertApp] CRITICAL | The 'export_format' trait of a
NbConvertApp instance must be any of ['custom', 'html', 'latex',
'markdown', 'python', 'rst', 'slides'] or None, but a value of
u'pdf' was specified.
```

Hopefully, this log could be easily clustered, because it's generic (doesn't contain any filesystem path, user information ect...).

Suppose there is 100 previously seen instances of this error, and 10'000 correct command execution with this parameter “--format=pdf”

In 95% of previously seen error instances, a package texlive-latex-extra was not installed.

In 98% of successful command executions, a package texlive-latex-extra was installed.

It's the only statistically significant difference between action routes of these two groups.

Based on that, the SLL algorithm will generate proposed solution as follows:

“sudo apt-get install texlive-latex-extra”

This is just a simplification of the algorithm. It can be extended in many ways.