IBM Mining Effectively Large Output Data Yield (MELODY)

Overview

Modern computer systems generate huge volumes of log files. Researchers at the IBM Lab in Haifa have developed an innovative system to handle these collections and sift out the "interesting" parts.

A typical log looks like this:

```
04/10/2010 12:34:56   INFO 123 123 123 123 123 123
04/10/2010 12:35:00   ERROR 123 123 123 123 123 123
04/10/2010 12:35:05   WARNING 123 123 123 123 123 123
04/10/2010 12:35:10   DEBUG 123 123 123 123 123 123
```

With so much information in each log, often we cannot see the forest for the trees. Finding the relevant parts requires intelligent processing, which is usually done with the help of specific rules. The challenge lies in applying these rules to a huge collection of log files.

Smarter Log Analysis

New machines are getting faster, larger, and more complex all the time. Reading their logs has become impossible for a human, and therefore some logs are "write only" and some are only read "post-mortem". Our new technology puts these logs to better use by highlighting the relevant sections.

The same technology can also be used to analyze system configuration files, QA logs, and more.

Advantages

- **Reduction of warranty costs** – quickly identifies the location of problems within the computer system and reduces the human effort required.

- **Improved processes** – helps detect problems early on and reduces the probability of ‘bogus’ problems that result, for example, in unnecessary part replacement.

- **Quality assurance** – sifts through log files to identify important messages generated during testing.

- **Root cause analysis** – finds the root cause of a complex problem by analyzing the logs generated.

- **Early detection** – helps detect errors during startup and prevent subsequent system crashes later on.

How It Works

MELODY works in three stages:

- **System deployment**: The system is integrated within the specific customer infrastructure and builds the initial machine learning model based on several months of real life data.

- **On-line analysis**: The incoming logs and configuration files are analyzed. This stage provides a concise human readable summary of the data.

- **Model retraining**: The model is routinely updated by learning new real data to ensure it continuously reflects the nature of the logs. This step is executed every few weeks.
Features

MELODY can work at several levels of implementation: from the simplest message level events, through meta-events at the highest level of system view.

Find rare events: The simplest version of MELODY learns the normal behavior of logs and/or data configuration parameters and then highlights any unusual data. This layer is being used within IBM to handle 9,000 log files a month from 3,000,000 different machines. The system provides two types of reports:

1. Log analysis: summarizes and highlights the important contents of the logs:

   ![Melody Log Analysis](image)

   - Drill down: This layer allows users to drill down to the message level. This feature is useful for finding problems in Q/A logs of manufactured machines, before they are shipped to the customers.

   - Clustering: Clustering the messages from different sources into meaningful groups gives a better sense of the ‘big picture’. This helps users understand the state of the system in terms of system-wide events.

   - Structured deployment: Software systems deployed in the field are upgraded periodically. The upgrade window is a time of vulnerability where problems are likely to occur. Misconfiguration, bugs, resource bottlenecks, and more; all conspire to occur during this time frame. Mining Effectively Large Output Data Yield (MELODY) can analyze changes between software upgrades and highlight anomalies across release boundaries.

2. Configuration analysis: detects rare events in the configuration details.

   ![Melody Device Analysis](image)

Unified timeline: This second layer presents the data using a common timeline, combining several data sources into a single holistic picture. This feature is especially useful when dealing with data coming from separate sources like operating system errors, database logs, application messages, and more.

Contact

Ehud Aharoni
Ofer Lavi
Ya’ara Goldschmidt, vaarag@il.ibm.com