Анализ бинарной совместимости репозитария rpm пакетов

Алексей Турбин
at@altlinux.org

Разработан метод анализа бинарной совместимости репозитария rpm пакетов, основанный на извлечении информации о динамических символах из исполняемых файлов и разделяемых библиотек формата ELF. Создана реляционная модель данных, в которой каждый динамический символ, ассоциированный с ELF файлом и rpm пакетом, является либо предоставляемым, либо требуемым. Предметом анализа является ссылочная целостность (разрешимость) символов, которая формализуется с помощью оператора соединения (`JOIN`). Символы, которые не удается таким образом разрешить, образуют некоторый класс потенциальных ошибок (бинарной несовместимости).

Семантически символы соответствуют функциям и глобальным переменным, используемым в приложении. Как правило, требуемые символы исполняемого файла должны быть сопоставлены с предоставляемым символам разделяемых библиотек, с которыми этот файл связан (разделяемые библиотеки в свою очередь могут быть связаны с другими разделяемыми библиотеками); невозможность сопоставить (разрешить) какой-либо требуемый символ приводит к аварийному останову приложения. Известно, на стадии компиляции (компоновки) исполняемого файла предусмотрены некоторые проверки разрешимости символов. Однако в ежедневно обновляемом репозитарии rpm пакетов этих проверок оказывается недостаточно, поскольку среда выполнения приложения может отличаться от среды, в которой собран соответствующий rpm пакет.

В первоначально разработанной упрощенной модели производится проверка разрешимости на полном множестве символов, без учета связей с разделяемыми библиотеками. В рамках этой модели можно обнаружить лишь некоторые подмножество потенциальных ошибок в разрешимости символов, поскольку считается, что любой требуемый символ может разрешиться в любой соответствующий ему предоставляемый символ. Тем не менее, при выполнение некоторых ограничений (в частности, при ограничениях на дублирование бинарного кода в репозитарии) в рамках модели удается получить практически значимые результаты.

Дублирование бинарного кода само по себе является достаточно актуальной проблемой. Из соображений переносимости некоторые приложения включают в себя внутренние копии библиотек, доступных в репозитарии в виде отдельных пакетов. Переносимость в таком случае противопоставляется модульной структуре и целостности репозитария: исправления в системной библиотеке оказываются недоступными приложениям.

В рамках рассмотренной модели разработан метод обнаружения дублирования бинарного кода, основанный на подсчете совпадающих символов, предоставляемых ELF файлами. Большое число совпадающих символов говорит о потенциальном

дублировании реализации какого-либо бинарного интерфейса. Этот метод также не является достаточным, поскольку формат ELF допускает сокрытие символов. Однако с помощью данного метода удалось, например, обнаружить в репозитарии несколько внутренних копий библиотеки `zlib`.

Дальнейшее развитие модели предполагает извлечение дополнительной информации из ELF файлов, в частности, связи с разделяемыми библиотеками (`DT_NEEDED`) и названия библиотек (`DT_SONAME`). В рамках новой модели удается обнаружить надмножество неразрешимых символов, поскольку разрешимость в этой модели не является транзитивной, тогда как динамический редактор связей допускает транзитивное сопоставление символов. Кроме того, в рамках этой модели можно ответить на следующие вопросы: 1) для данного ELF файла, существуют ли символы, которые разрешаются в несколько разделяемых библиотек, с которыми этот файл связан? множество таких символов образует ещё один класс потенциальных ошибок; 2) среди библиотек, с которыми связан ELF файл, существуют ли такие, которые он не использует? последнее обнаруживает, в частности, излишние и чрезмерно жесткие зависимости между пакетами. Показано, что в среднем почти половина всех библиотек, с которыми связаны ELF файлы, образуют излишние связи. Впоследствии эта проблема была в основном решена с помощью включения по умолчанию специального режима компоновки `ld --as-needed`.

Наконец, разработан метод анализа бинарной совместимости между двумя «срезами» репозитария, сделанными в разное время. Предметом анализа является возможность частичного обновления системы, построенной на основе «старого» репозитария, пакетами из «нового» репозитария, без нарушения совместимости между бинарными интерфейсами. В этой задаче символы, предоставляемые разделяемыми библиотеками, образуют отдельное отношение; производится сравнение разрешимости символов на множестве библиотек «старого» и «нового» репозитария. Если разрешимость невозможна в одном из случаев, но не в другом, это говорит о недопустимом изменении бинарного интерфейса соответствующей разделяемой библиотеки.

Рассмотрена реализация предложенных моделей, в которой отношения представлены в виде текстовых файлов, а операции реляционной алгебры реализованы с помощью стандартных утилит UNIX. Утверждается, что ни один доступный SQL сервер не способен обеспечить приемлемую производительность для решения рассмотренных задач.

Список литературы

[1] Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification Version 1.2. `http://www.x86.org/ftp/manuals/tools/elf.pdf`

[2] Ulrich Drepper. *How To Write Shared Libraries.* `http://people.redhat.com/drepper/dsohowto.pdf`

[3] E. F. Codd. *A Relational Model of Data for Large Shared Data Banks.* Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377–387. `http://www.acm.org/classics/nov95/`

[4] Evan Schaffer and Mike Wolf. *The UNIX Shell as a Fourth Generation Language.* `http://www.rdb.com/lib/4gl.pdf`