

Технологии и принципы создания универсального журнала операций

77-48211/640626

10, октябрь 2013

Белоусова В. И.

УДК 004.457

Россия, МГТУ им. Н.Э. Баумана

belousova.vi@gmail.com

Введение

В сложных распределённых системах одновременно выполняется множество операций, и в случае возникновения неисправности или отказа того или иного модуля системы бывает достаточно сложно сразу же определить причину неисправности или отказа. Поэтому важно наличие централизованного механизма получения и хранения сообщений и протоколов о работе программных модулей и подсистем в составе любого программного комплекса [1].

Каждый модуль системы, выполнив ту или иную операцию, должен добавить запись в соответствующий журнал операций. Использование журналов операций сокращает время на поиск причин возникающих неисправностей [2], так как становится возможным однозначно определить, после выполнения какой операции произошёл отказ.

Поскольку комплексы программ независимо от своей прикладной сферы должны иметь средства протоколирования операций [3, 4], то целесообразно создать универсальный компонент для ведения журнала операций с возможностью подключать его к различным программам.

1 Встроенные средства ведения журналов

Прежде всего следует рассмотреть встроенные механизмы журнализации, используемые в операционных системах, на предмет эффективности использования их для протоколирования операций.

В операционной системе Microsoft Windows используется внутренний механизм журнализации – EventLog. Кроме локального ведения журнала есть возможность отправки сообщений в удалённый журнал, расположенный на другой машине в сети. Недостатком

является ограничение использованием только в операционных системах семейства Microsoft Windows, что не подходит для кроссплатформенной разработки [5, 6, 7].

В Unix-подобных операционных системах используется стандарт журнализации syslog. Имеется возможность ведения удалённого журнала, то есть отправлять сообщения по сети на удалённый сервер. К недостаткам данного подхода относятся использование ненадёжного протокола UDP, возможность подмены сообщений или отправка пустых сообщений и последующая перегрузка сервера, а также отсутствие поддержки в операционных системах Microsoft Windows [8, 9].

Следовательно, встроенные механизмы журнализации являются недостаточно эффективными для их использования в журнале операций по причине зависимости от платформы.

2 Требования к журналу операций

Сформулируем основные требования к программному комплексу журналирования, которые позволят использовать его в различных программах. Для обеспечения универсальности интерфейсов доступа Журнал должен получать сообщения от других программных комплексов следующими способами:

- синхронный вызов прикладных функций внутри исполняемого потока;
- асинхронные сообщения от внешних процессов/или потоков;
- передача сообщения через утилиту;
- обращение к разделу базы данных журнала как вызов хранимой процедуры или запрос к представлению.

Хранение данных журнала в базе данных обусловлено тем, что это позволит обеспечить журнализацию операций на уровне БД без вызова внешней программы; выполнять быстрый поиск и фильтрацию записей журнала; обеспечить параллельную запись в журнал нескольких приложений.

Необходимо предусмотреть возможность получения сообщений в различных форматах с последующим преобразованием к единому формату. Единый формат должен предоставлять возможность хранения дополнительных сведений о сообщении для обеспечения адаптации под нужды программных комплексов.

Также журнал должен быть кроссплатформенным приложением.

3 Формат хранения сообщений в журнале

Различные программные комплексы могут вести собственные журналы выполняемых операций, а также журналы ошибок. Рассмотрим различные способы журнализации и трудности доступа к ним.

Данные журнала хранятся в файле. Файл может быть как текстовым, так и собственного специфичного формата, понятного только этому программному комплексу. Однако, даже в случае использования обычного текстового файла информация может быть организована в нём по какой-то своей структуре.

Данные журнала хранятся в собственной базе данных. В этом случае потребуется устанавливать соединение с БД этого программного комплекса, а значит, необходимо знать сервер, порт, имя пользователя, пароль и, что особенно важно, структуру БД. На установку соединения и извлечение данных из БД требуется время. Кроме того, сервер с внешней БД просто может оказаться недоступен по тем или иным причинам.

Таким образом, каждый программный комплекс может иметь свой собственный формат ведения журнала операций и ошибок [10]. Чтобы получать данные из каждого собственного журнала необходимо уметь работать с его форматом и знать его структуру. Недостатки такого подхода очевидны – невозможно учесть все существующие форматы журналов. Особенно с учётом того, что в будущем может потребоваться работать с новыми, до этого неизвестными форматами журналов.

Поэтому решением проблемы является разработка собственного формата сообщений журнала, в который будут преобразовываться все поступающие сообщения. Собственный формат сообщения подразумевает набор некоторых обязательных полей, а также дополнительные поля для хранения информации, которая не может быть приведена к общему формату. Для отсутствующих полей предусматриваются значения по умолчанию.

На основе обобщенных требований к данным, содержащимся в журнале операций, предлагается следующий состав полей сообщений:

- текст сообщения;
- статус сообщения;
- код сообщения;
- идентификатор модуля-отправителя сообщения;
- имя пользователя-отправителя сообщения;
- IP-адрес рабочей станции пользователя-отправителя;
- дата и время отправки сообщения.

Такой набор полей будет считаться единым форматом сообщения. Если во входящем сообщении отсутствуют значения каких-либо полей, то этим полям присваиваются значения по умолчанию, например, дата и время, имя пользователя, IP-адрес.

Кроме обязательных полей необходимо предусмотреть возможность хранения дополнительных полей сообщения, состав которых определяется нуждами конкретного

программного комплекса. Сообщения для добавления записей в раздел БД журнала могут поступать в различных форматах. Возможны следующие ситуации:

- сообщение поступило в формате записи журнала. В этом случае оно уже готово к добавлению в Журнал;

- сообщение поступило в формате отличным от формата журнала, и набора данных не хватает для заполнения всех полей. Тогда вместо недостающих данных необходимо будет подставлять значения по умолчанию или генерировать их автоматически;

- сообщение поступило в формате, отличным от формата журнала, и набор данных избыточен. В этом случае следует заполнить обязательные поля и сохранить исходное сообщение в дополнительном поле. Для хранения исходного сообщения предлагается использовать формат XML;

- сообщение поступило файлом не текстового формата, например, изображение или звукозапись. Тогда следует добавить файл в архив и сохранить в дополнительном поле ссылку на запись из архива;

- сообщение уже хранится в другой, сторонней БД и нужно указать ссылку на него. Следует учесть, что просто дать ссылку на запись не получится, так как необходимо знать сервер этой БД, данные аутентификации, сведения о внутренней структуре БД и т.д. Все эти данные также предлагается хранить в дополнительном поле в формате XML.

Итого получаем следующий набор дополнительных полей:

- поле для хранения в формате XML оригинала сообщения, набор данных которого избыточен относительно формата сообщений журнала;

- поле для хранения ссылки на файл или на группу файлов;

- поле для хранения в формате XML сведений о записи сторонней БД, в которой содержится сообщение;

- поле для указания исходного формата сообщения.

Указание исходного формата сообщения поможет упростить запросы на извлечение записей определенного формата и их разбор.

Дополнительные сведения о сообщении удобнее всего хранить в формате XML [11], поскольку этот формат легко расширяем; содержит в себе как структуру, так и данные; допускает альтернативность структуры данных; позволяет не изменять существующие анализаторы сообщений при расширении структуры XML за счет игнорирования части полей.

4 Компоненты журнала операций

Журнал операций должен быть реализован как набор прикладных функций, обеспечивающих получение сообщений от программных комплексов, преобразование получаемых сообщений в единый формат и сохранение последних в разделе БД журнала. В состав программного комплекса должны входить:

- API Журнала. API представляет собой набор функций, которые вызываются другим процессом, принимают на вход набор параметров, выполняются и возвращают результат;

- утилита Журнала. Утилита представляет собой консольное приложение (без графического интерфейса пользователя), которое запускается другим процессом или пользователем (оператором, администратором), принимает параметры из командной строки, вызывает функции API Журнала и завершается;

- приложение с графическим интерфейсом. Приложение с графическим интерфейсом пользователя (GUI) предназначено для администратора и обеспечивает возможность просмотра, экспорта и удаления записей из журнала. Просматривать можно как все записи, так и осуществлять выборку с параметрами.

Такое разбиение Журнала на компоненты обеспечивает, во-первых, отсутствие дублирования функций и легкость модернизации программы Журнала за счет локализации функционала в API. Во-вторых, позволяет добавлять сообщения из программ, написанных на различных языках программирования, удаленно по сети и даже вручную оператором, что существенно расширяет область применения Журнала. И наконец, предоставляет удобный интерфейс администратору для поиска и анализа возникших проблем в работе системы.

API Журнала должен обеспечивать выполнение следующих функций: добавление новых записей в БД Журнала; просмотр существующих записей в БД Журнала; экспорт записей из БД Журнала в файл.

Добавление записей в Журнал должно поддерживаться в двух вариантах: в стандартном формате и в формате XML. Добавление стандартного сообщения выполняется функцией API Журнала, которая на вход принимает объект сообщения с полностью или частично заполненными полями. Добавление сообщения в формате XML позволит сформировать из него сообщение стандартного формата, а также сохранить оригинал в БД Журнала.

Функции API вызываются всеми модулями системы, которые будут добавлять записи в Журнал. Вызов функций API должен выполняться из утилиты, работающей в режиме командной строки; в синхронном режиме из функций внутри исполняемого процесса; в

асинхронном режиме в виде сообщений от другого процесса. В последнем случае Журнал должен быть реализован как служба.

Утилита запускается вручную пользователем (оператором, администратором) или другим процессом. Для взаимодействия с пользователем утилита имеет интерфейс командной строки с возможностью вывода сообщений в окно консоли (терминала). Утилита использует функции API Журнала.

Сообщение для добавления в Журнал через аргументы командной строки может быть передано следующими способами:

- передача всего XML-сообщения одной строкой;
- передача только имени XML-файла (и пути к нему).

При передаче XML-сообщения одной строкой возникает проблема длины строки. В ОС Linux длина строки аргументов составляет 64 КБ. Кроме того, стандартная утилита `xargs` позволяет разбивать строку аргументов на несколько. Таким образом, для передачи XML-сообщения в качестве параметра запуска данного объёма должно быть достаточно. Однако, в ОС Microsoft Windows XP и более старших версиях длина командной строки ограничена 8191 символом, а в Microsoft Windows 2000 и Windows NT 4.0 - 2047 символов, а в более ранних версиях на длину командной строки наложено ограничение в 128 символов. Все эти ограничения необходимо учитывать при выборе ОС, в среде которой будет производиться эксплуатация программ [8].

Однако, передача сообщения одной строкой может быть удобнее с точки зрения использования или по мнению эксперта [12]. Поэтому предлагается использовать комбинированный вариант, поддерживающий обе возможности.

В этом случае формат запуска утилиты будет иметь вид:

`<утилита> <-str | -file> <"строка">`

Значения набора параметров:

- `str` показывает, что далее следует строка XML-сообщения;
- `file` показывает, что далее следует имя файла.

В зависимости от указанного параметра строка будет обрабатываться соответствующим образом.

Графический интерфейс обеспечивает визуальное представление таблицы записей журнала. В базовом варианте он должен поддерживать возможность поиска и фильтрации записей, просмотра деталей сообщения, экспорт старых записей журнала в архивные файлы с возможностью их последующего просмотра.

При просмотре деталей сообщения необходимо предусмотреть просмотр оригинала в формате XML; извлечение и открытие файлов, связанных с сообщением в соответствующих приложениях; обращение к внешней БД за дополнительными сведениями о сообщении. В последнем случае установка соединения с удаленной БД должна проводиться в соответствии с данными, хранящимися в дополнительных полях сообщения.

Графическое приложение запускается вручную администратором и использует в своей работе функции API Журнала.

Заключение

Для создания централизованного и универсального журнала операций, способного работать в кроссплатформенной среде, был проведен анализ встроенных средств журнализации операционных систем, который показал ограниченность их использования. Поэтому были определены требования к журналу операций и разработаны основные концепции по его созданию.

Предложен единый формат сообщений журнала, который содержит типовые поля сообщения и позволяет хранить сообщения в других форматах, в том числе в виде файлов или таблиц внешних БД. Разработанный формат обеспечивает единое представление сообщений и одновременно возможность хранения произвольного набора данных. Разработана логическая структура программы журнала операций, обеспечивающая оптимальный набор компонентов для работы в кроссплатформенной среде с возможностью удаленного доступа. Предложен набор функций API, утилиты и графического интерфейса журнала, которые обеспечивают базовый функционал типового журнала.

Предложенные принципы и технологии создания журнала операций являются универсальными и могут быть применены при разработке журналов операций во многих программных комплексах.

Список литературы

1. Черненко В.М., Гапанюк Ю.Е., Мавзютов А.А. Разработка комплексных биомедицинских информационных систем на основе адаптивных объектов // Вестник МГТУ им. Н.Э. Баумана. Серия "Приборостроение", 2012. Спец.выпуск № 3 "Биометрические технологии". С. 105 — 112.
2. Виноградова М.В., Гжельская М.О. Технология проектирования баз данных на примере пропускной системы общежития // Проблемы построения и эксплуатации систем обработки информации и управления: Сб. статей (М.) под ред. Черненко В.М.. – 2011. Выпуск 8.

3. Большаков С. А. Реализация и опыт применения персонального сайта преподавателя // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 05 Режим доступа: <http://technomag.bmstu.ru/doc/585850.html> (Дата обращения 05.10.2013)
4. Виноградов В.И., Мазнев В.Г. Метод поиска часто повторяющихся маршрутов в пространственно-временных данных // Вестник МГТУ им. Н.Э. Баумана. Серия "Приборостроение" 2012 . Спец.выпуск №5 "Информатика и системы управления". С. 13 - 19
5. Справочная информация Microsoft по журналам событий. Режим доступа: <http://technet.microsoft.com/ru-ru/library/cc722404.aspx> (Дата обращения 05.10.2013)
6. Вернигора А. Журналирование в Windows // Windows IT Pro. Издательство «Открытые системы». Электрон. Журнал. 2011. № 02. Режим доступа: <http://www.osp.ru/win2000/2011/02/13008443/> (Дата обращения 05.10.2013)
7. Просмотр событий в Windows 7. Режим доступа: http://www.oszone.net/10807/http_oszone_net_10751_event_viewer_3 (Дата обращения 05.10.2013)
8. Немец Э., Снайдер Г., Хейн Т., Уэйли Б. Unix и Linux. Руководство системного администратора. М.: Вильямс, 2012. 1312 с.
9. Системные журналы Linux (управление логированием). Режим доступа: <http://www.k-max.name/linux/syslogd-and-logrotate/> (Дата обращения 05.10.2013)
10. Белоусова В.И., Игушева Ю.С. Классификация источников данных и создание унифицированной модели представления данных / Проблемы построения и эксплуатации систем обработки информации и управления: Сб. статей под ред. Черненко В.М. - М.: Издательство МГТУ им. Н.Э. Баумана. – 2011 г., Выпуск 9. - с. 44-56.
11. Официальная спецификация стандарта XML 1.0. Режим доступа: <http://www.w3.org/TR/REC-xml/> (Дата обращения 05.10.2013)
12. Постников В. М., Спиридонов С. Б. Подход к увеличению уровня согласованности мнений экспертов при выборе варианта развития системы обработки информации // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 06 Режим доступа: <http://technomag.bmstu.ru/doc/574220.html> (Дата обращения 05.10.2013)