УДК 004.052.32, 004.052.42, 004.418

Н.В. Злобина, Н.В. Волжанкин, Н.Е. Пособилов

ОБЕСПЕЧЕНИЕ ЦЕНТРАЛИЗОВАННОГО МОНИТОРИНГА ДЛЯ СИСТЕМ СЛОЖНОЙ АРХИТЕКТУРЫ С БОЛЬШИМ ОБЪЁМОМ ДАННЫХ

Нижегородский государственный технический университет им. Р.Е. Алексеева

Рассматривается проблема централизованного управления логами процессов. Выдвинуты основные требования к внедряемой системе, обозначена главная цель системы и обозначен список задач, которые должна решать система для улучшения качества и эффективности работы ИТ отдела. Проведен сравнительный анализ существующих решений. Рассмотрены основные особенности и характеристики лучшего решения, которое впоследствии было внедрено на предприятии.

Ключевые слова: мониторинг, логирование, IT-процессы.

На предприятии существует большое количество сложноконтролируемых процессов (приложений), распределенных по серверам локальной сети и персональным рабочим станциям. Эти процессы (приложения) логируют свои активности в текстовые файлы в разных объёмах и с разной скоростью (рис. 1).

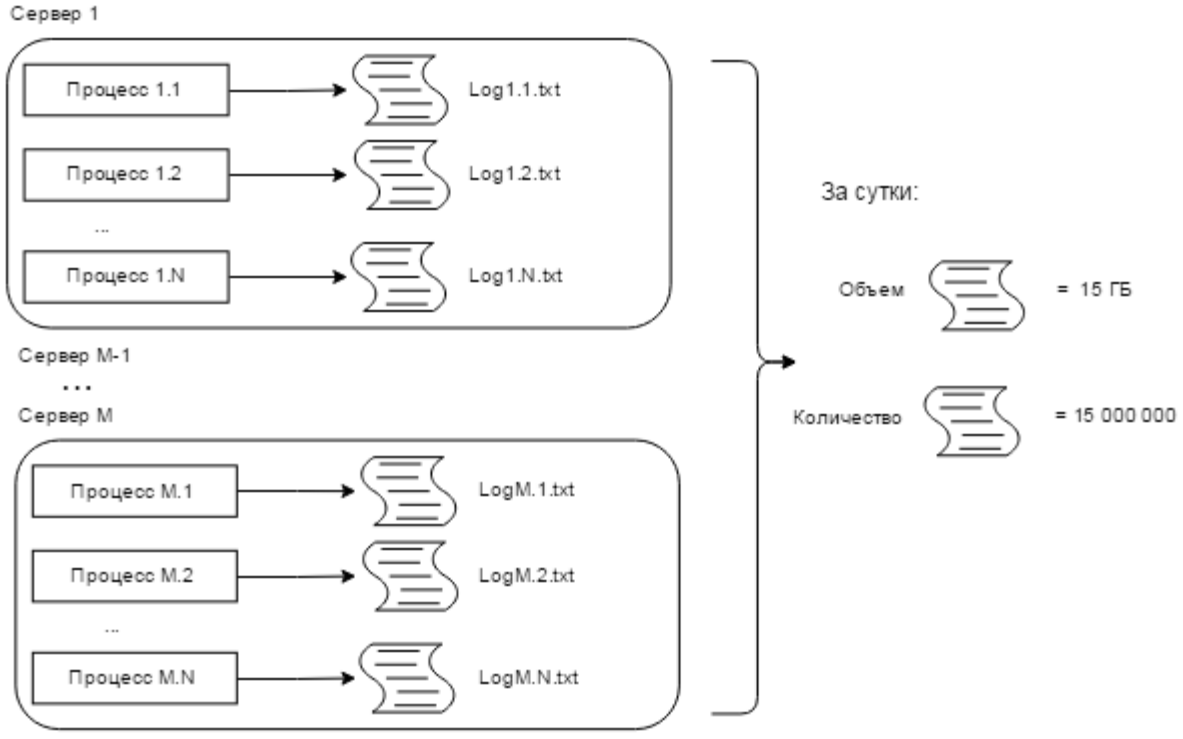

Рис. 1. Схема процессов на предприятии

При такой организации рабочего процесса выявлены следующие проблемы:

- текстовые логи занимают большое количество места на серверах;
- поиск информации о логах процесса занимает крайне много времени;
- узнать о проблеме, ошибке или крахе приложения своевременно с помощью логов практически невозможно.

Чтобы иметь представление о результатах и эффективности работы приложений на

© Волжанкин Н. В., Злобина Н. В., Пособилов Н. Е., 2017.

серверах, необходимо заниматься изучением файлов с логами, на что требуется очень много времени. При этом любое нестандартное поведение может повлечь за собой отказ того или иного процесса или их комплекса. В таких условиях обнаружить проблему на одном из многочисленных серверов среди взаимосвязанных процессов сложно и трудоёмко.

Таким образом, выявлены неэффективности по показателю времени работ, которые во всех случаях влекут за собой убытки компании.

Чтобы избежать ситуаций, приводящих к убыткам, когда первыми ошибки в системе замечают пользователи продукта, а не сотрудники предприятия, необходимо:

- контролировать процессы;
- вести статистику работы процессов;
- знать о любых отклонениях от нормы в поведении процессов;
- получать уведомления о нештатных ситуациях;
- узнать слабые стороны процессов и понять, в каком направлении вести доработки и оптимизацию.

Поэтому необходимо проводить исследования по возможности автоматизации контроля и работы с текстовыми логами путем внедрения централизованной системы мониторинга.

Целью является исследование способов решения проблем в работе с текстовыми логами на предприятии и дальнейшая оптимизация работы путем внедрения централизованной системы мониторинга.

Глобальный мониторинг по локальной сети, в состав которой входят серверы, точки доступа, персональные рабочие станции и прочее сетевое оборудование, не требуется – это задача системных администраторов. Текущая проблема ограничивается одним отделом, внутри которого необходимо контролировать лишь выборочные процессы и программы, для этого необходима система централизованного сбора, хранения и анализа информации из лог-файлов.

Для организации системы контроля и слежения за процессами составлен следующий алгоритм действий:

- собрать информацию о процессах логирования, существующих решениях обозначенной проблемы;
- провести сравнительный анализ всех вариантов и выбрать лучший;
- организовать работу программ и приложений так, чтобы их результаты были измеряемыми;
- внедрить систему мониторинга;
- провести анализ нового рабочего процесса и его результатов;
- обучить специалистов тому, как проводить мониторинг и оценку;
- встроить мониторинг и оценку в деятельность организации.

Запись логов необходима по нескольким причинам:

- появляется возможность узнать всё о состоянии системы в настоящий момент без помощи отладчика;
- доступна информация о прошлых состояниях системы, что позволяет узнать причины, по которым система оказалась в определенном состоянии;
- есть возможность проведения анализа потребления ресурсов.

Логи по признаку действующего лица можно разделить на два типа: действия и события с точки зрения пользователя и с точки зрения разработчика.

По области, к которой относятся логи, можно разделить на три типа: системные (SYSTEM) , безопасности (SECURITY) и приложения (APPLICATION). Например, таким образом классифицируются системные события в Windows семейства NT.

Процесс логирования необходимо реализовывать, основываясь на следующих правилах:

- максимально короткие сообщения;
- максимально информативные сообщения;
- вывод лога в отдельной строке;
- локализация ошибки только для редких сообщений;
- лучше не использовать преобразования типов;
- разделение по уровням логирования.

Уровни помогают определить критичность сообщения и приемлемое время реакции на него. Далее приведены допустимые уровни журнала (в порядке убывания критичности): Fatal, Error, Warn, Info, Debug, Trace.

Ротация файлов

Со временем количество файлов и их вес могут стать критичным для системы, поэтому необходимо при наступлении определенных условий необходимо подменять активный файл записи.

Процесс ротации:

- архивация старого лога;
- сохранение с другим именем, например: log.txt (текущий файл) сохраняется как log1.txt;
- очистка текущего файла логирования;
- удаление старых логов по определенному признаку;
- возможный перезапуск процесса записи логов.

Чаще всего используемые условия: привязка к дате (регулярная ротация) или размеру файла (ротация по достижению критического размера).

Асинхронная запись логов

Сообщения некоторого типа можно записывать пачками, а более критичные сообщения требуют мгновенной записи, но необходимо, чтобы размер буфера был гибко настраиваемый.

Когда производится асинхронная запись лог, возможны три варианта поведения.

1. Если очередь не заполнена, регистрационное сообщение записывается. В противном случае, если очередь полна (количество сообщений в нем достигается max_size), то вызов блокируется до тех пор, пока очередь не закончится.
2. Рабочий поток выталкивает сообщение из очереди и регистрирует сообщение, в зависимости от уровня лога.
3. Если исключение происходит в рабочем потоке (например, не удалось войти в файл) исключение будет повторно сгенерировано, когда пользователь вызовет следующее сообщение журнала. Таким образом, пользователь может получать уведомления об ошибках, происходящих в рабочем потоке.

Конфигурация файлов

Ответа на вопрос, сколько нужно логировать информации, – нет. С точки зрения конечного пользователя – чем меньше, тем лучше. Меньше тратится ресурсов и процессорного времени на запись и хранение и больше времени остается на работу. Но с точки зрения разработчиков, количество логов должно быть максимальным, чтобы в любом случае получать полный отчет о работе приложения. Чтобы определиться с количеством и качеством логов для конкретного случая, необходимо часто менять и настраивать процесс логирования. Для удобства в работе этого процесса необходимо отдельно вынести его настройки. Необходимо, чтобы такая настройка была отделена от приложения, таким образом реализуется возможность изменения конфигураций без изменения программного кода. (например в *NIX - сервис syslog, в Windows семейства NT – NTEventViewer.) Таким образом, получаем возможность гибкой настройки места логирования или сообщений определенного типа.

Проверка лог-сообщений

Если зарегистрированная информация лога является частью фундаментального бизнес-процесса, то было бы правильно добавить некоторые тесты для проверки. Необязательно проверять каждое лог-сообщение и каждую функцию, но функции, являющиеся основополагающими в бизнес-процессе, выполняющие критически важные задачи, требуют больше внимания и контроля со стороны разработчиков. Например, функция оплаты заказа в интернет-магазине явно имеет высокую критичность. Поэтому при поступлении лог-сообщения об ошибке в процессе оплаты необходимо удостовериться, что такая ошибка действительно существует и может привести к серьезным убыткам. Далее представлен алгоритм тестирования логгеров:

1. Получить ссылку на тестируемый логгер.
2. Сохранить уровень ведения журнала для этого регистратора (для восстановления после запуска теста).
3. Поднять уровень ведения журнала до ожидаемого уровня.
4. Выполнить код, который должен запускать ведение журнала.
5. Проверить правильность сообщения и информации.
6. (После теста - то есть срыва) восстановить уровень ведения журнала до исходного значения.

В результате проведенных исследований было найдено несколько доступных решений, для обозначенной проблемы. Далее описаны особенности каждой из систем.

Logstash. Бесплатное OpenSource приложение отвечает за прием логов по сети, фильтрацию, категоризацию, парсинг и дальнейшую передачу elasticsearch. Система в целом представляет из себя связку:

- elasticsearch – поисковый движок. Обрабатывает логи, что в дальнейшем обеспечивает очень быстрый поиск архивным логам;
- kibana – веб-приложение с удобным интерфейсом для визуализации и фильтрации логов;
- logstash-forwarder – агент на сервера который будет отправлять нужные логи на демон logstash.

Есть возможность использовать фильтры для сбора логов приложений и структурирования данных, благодаря чему данные можно легко запросить и проанализировать.

Недостатки: необходимость Java на всех серверах, высокие требования к памяти, потенциально может влиять на работу других сервисов, сложный в настройке.

Splunk. Входными данными являются произвольные текстовые данные, разбитые на строки. Способы их доставки в Splunk весьма гибкие. Самое простое – установить на сервера агент Splunk, указать ему каталоги или конкретные файлы, которые надо мониторить, и он будет автоматически отсылать обновления на центральный сервер. Можно самостоятельно доставлять логи на сервер Splunk, например, посылая их туда через по HTTP, TCP, UDP. ЕщеSplunkумеетбратьданныеиз Windows (Logs, Events, Performance Counters).

Недостатки: лицензия SplunkFree предназначена для индивидуального использования, есть ограничение на максимальный объем индексации в 500 МБ/день.

Graylog. Бесплатное OpenSource приложение ставится в связке с elasticsearch, имеет свою систему визуализации, систему сбора логов. Организованный сбор событий, фильтрация, поиск, автоматизация, аналитика и интерфейс. Состоит из следующих компонентов:

- graylog2-webui — web-интерфейс на Rails;
- graylog2-server — Java TCP/UDP лог-коллектор;
- mongodb для хранения и настроек всей системы в целом.

Недостатки: плоская схема базы.

Проведенный сравнительный анализ показал, что большими преимуществами в заданных условиях обладает система GrayLog. Основными преимуществами системы являются:

- аггрегация сообщений в streams. Объединение потоков логов с нескольких хостов по ключевому слову;
- на stream можно настроить оповещения и отправлять их на почту;
- аггрегация хостов в группы. Можно объединить потоки с разных хостов в одну группу;
- выборки из всего массива по regexp, по времени, по важности, по facility;
- блеклисты для логов. По regexp можно отфильтровать логи. Возможность настроить запрет на запись в базу для некоторых потоков;
- авторотация логов. Вычищение старых записей на mongodb, которая автоматически работает с помощью механизма «cappedcollections»;
- возможность использования GELF — graylogextendedlogformat, расширяя.стандартную длину syslog сообщения в 1024 байта. С помощью GELF можно отслеживать не только системные сообщения, но уже и логику работы кода, посылая развернутые сообщения прямо из приложения;
- проект бесплатный (GPLv3), активно развивается.

Для осуществления централизованного управления и мониторинга были развернуты системы на базе свободно распространяемых продуктов, что обеспечило максимально эффективное решение. Схема внедренной системы представлена на рис. 2.

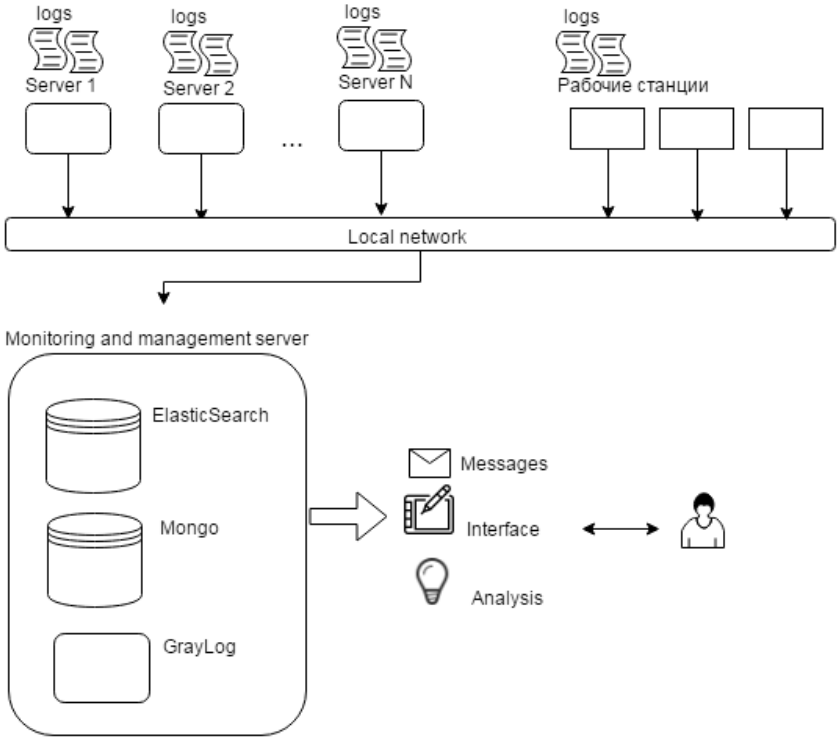

Рис. 2. Схема внедрения системы

Результаты

На одном из IT предприятии, внедрив централизованную систему GrayLog для работы с логами, удалось автоматизировать следующие процессы:

- мониторинг работоспособности и производительности процессов в рамках IT отдела;
- оповещения о проблемах технических специалистов, ответственных за соответствующие компоненты информационной инфраструктуры до того, как они нанесли существенный ущерб и повлияли на работу пользователей;
- анализ и визуализация статистических данных.

Также в результате внедрения системы удалось повысить эффективность труда сотрудников информационных отделов за счет снижения трудозатрат на поиск проблем и их причин, а также автоматизации ряда рутинных операций;

Библиографическийсписок

1. **Олифер, В.** Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В. Олифер, Нат. Олифер. – 4-е изд. – СПб.: Питер, 2011. – 944 с.
2. Технологии анализа данных: DataMining, VisualMining, TextMining, OLAP : учеб. пособие / А.А. Барсегян [и др.]. – 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2007 .— 382 с.
3. **Лукас, М.** FreeBSD. Подробное руководство: [пер. с англ.] / М. Лукас. – 2-е изд. – СПб.: Символ Плюс, 2009. – 864 с.
4. Event Logs and Multithreaded Components[Электронныйресурс] // microsoft.comURL: https://msdn.microsoft.com/en-us/library/0680sfkd.aspx
5. Application logging principles[Электронныйресурс] // thekua.com, 2008 URL:https://www.thekua.com/atwork/2008/11/application-logging-principles
6. **Скрипник, В.М.** Оценка надежности ехнических систем по цензурированным выборкам / В.М. Скрипник, А.Е. Назин. – Минск: Наука и техника, 1981. – 144 с.
7. Ведение лога приложения[Электронный ресурс] // skipy.ru ,2010 URL: http://www.skipy.ru/useful/logging.html
8. .NET Logging tools and libraries [Электронныйресурс] // dotnetlogging.com, URL: http://www.dotnetlogging.com/comparison/
9. Анализ логов в реальном времени [Электронный ресурс] // habrahabr.ru, 2012, URL: https://habrahabr.ru/post/150657/
10. **Ромашко, Б.** Логирование проекта с помощью NLogFramework[Электронный ресурс] // itvdn.com, itvdn.comURL: https://itvdn.com/ru/blog/article/logging-project-with-nlog-framework

Дата поступления
в редакцию 15.08.2017

N.V. Zlobina, N.V. Volzhankin, N.E. Posobilov

PROVIDING CENTRALIZED MONITORING OF APPLICATIONS OF DISTRIBUTED INFORMATION SYSTEMS

Nizhny Novgorod state technical university n.a. R.E. Alekseev

Purpose: The goal is to optimizing the work with text logs, by implementing a centralized management system.
Design/methodology/approach: The scheme for implementing centralized management and monitoring is presented. For implementation, no funds are required, the system can be deployed on the basis of freely distributed products, which provides the most effective solution.
Findings: As a result of this work, information was collected on logging processes, existing solutions to the problem, a comparative analysis of all options was made and the best one was selected. The work of programs and applications is organized in such a way that their results are measurable, and a monitoring system is introduced. The analysis of a new workflow and its results is carried out.
Research limitations/implications: This monitoring system may have problems with the speed of processing and issuing information when the permissible volumes of incoming data increase with the indicated resources.
Originality/value: The analysis showed the most optimal solution to the problem. Having provided automated monitoring, it was possible to achieve an increase in the IT department performance indicators.

Key words: monitoring, logging, IT-processes.