Разработка многопроцессных и многопоточных приложений с доступом к общим ресурсам

Лабораторная работа

Ревизия: 0.1

Последняя модификация: Vlad Kovtun

Дата последней модификации: 06.03.2010 17:16:00

История изменений

03.03.2010 – Версия 0.1. Первичный документ. Влад Ковтун

Содержание

Лабораторная работа 3. Разработка многопроцессных и многопоточных приложений с доступом к общим ресурсам

Вопросы

1. Постановка задачи.
2. Методические рекомендации.

Постановка задачи

Цель

Разработать консольное приложение, которое анализирует логии web-сервера Apache с использованием параллельных вычислений и на выходе формирует заданное количество файлов логов web-сервера Apache, в которых упорядочены строки по возрастанию IP адреса.

Задачи

1. Ознакомиться с понятиями процессов и потоков в ОС.

2. Ознакомиться с понятиями межпроцессного и межпоточного взаимодействия.

3. Ознакомиться с понятиями совместного доступа к общим ресурсам, взаимоблокировками.

4. Разработать консольное приложение для параллельного анализа нескольких файлов логов web-сервера Apache.

4.1. На основе файлов логов web-сервера, следует сформировать:

- Заданное количество таких же файлов логов, но в которых записи из заданных файлов, распределены по IP адресам между заданным количеством результирующих файлов.
- Заданное количество таких же файлов логов, но в которых записи из заданных файлов, распределены по датам записей между заданным количеством результирующих файлов.

5. Разработать тестовое приложение, которое формирует все возможные тестовые комбинации ключей командной строки для приложения.

6.1. Оценить производительность приложения, при различном количестве поток чтения и записи. Количество потоков следует изменять от 1 до 8.

6.2. Оценить производительность приложения, при различном количестве процессов чтения и записи. Количество процессов следует изменять от 1 до 8.

7. Оформить функциональную спецификацию на приложение. В функциональной спецификации обязательно указать, каким образом производится тестирование на корректность.

8. Оформить отчет к лабораторной работе.

Вход

Для управления приложением (первый подход) предлагается использовать ключи командной строки:

- /id:<filedir> – полный путь к директории, в которой хранятся файлы логов для анализа. Данный параметр является обязательным, если он не указан, то происходит вывод на экран соответствующего сообщения и подсказки по использованию данного приложения.
- /if:<filenames> - перечисление имен файлов логов через пробел, которые необходимо проанализировать. Файлы обязаны находится в директории указанной с помощью ключа /id. Данный ключ позволяет осуществлять выборочный анализ файлов логов.
- /od:<filedir> – полный путь к директории, в которой будут храниться результирующие файлы логов. Данный параметр является не обязательным,

если он не указан, то формирование результирующих файлов происходит в текущую директорию.

- /ok:<count> - количество результирующих файлов логов. Данный параметр является не обязательным. По умолчанию количество результирующих файлов 3.
- /o:<filename%.ext> – шаблон имени файла, в которые будут заноситься результирующие файлы логов. Данный параметр является не обязательным, если он не указан, то формирование результирующих файлов происходит в файлы типа report%N%.dat.
- /thr:<threads count> - количество потоков, которые будут производить чтение данных из файлов и их запись в результирующие файлы. По умолчанию используется 2 потока.
- /? - вывод информации о допустимых ключах командной строки.

Пример

Подход 1. Анализ 3 файлов логов web-сервера Apache, результат анализа выводится в 5 файлов вида report_.dat.

C:/>analyser.exe /id:"c:/logs/" /if access-www.1.log access-www.2.log access-www.3.log /ok:5 /o:report_%_.dat

Вывод

Во время работы приложения, рекомендуется выводить информацию о статусе приложения, а также о корректности его работы на консоль.

Другими словами, необходимо анализировать, сколько процентов из каждого файла – проанализировано.

Допускается перенаправление вывода консоли в текстовый файл, например:

C:/>analyser.exe /id:"c:/logs/" /if access-www.1.log access-www.2.log access-www.3.log /ok:5 /o:statistics%.dat >report.txt

Требования

- Архитектура приложения строится по модульному принципу.
- За основу принимается стандартная библиотека C++ (в случае разработки на C++).
- Рекомендуется использовать защищенные ресурсы и указатели.
- Обязательным является обработка исключений.
- Во время работы приложения обязательным является отображение информации о статусе приложения, т.е. оно работает, обработано столько-то процентов данных.
- Следует предусмотреть возможность работы приложения в отладочном режиме, когда вся информация заносится в лог приложения. Допускается ведение персонального лога для каждого из процессов (потоков).
- Исходный код обязан быть комментирован. Для C# следует использовать нотацию Microsoft [1], для C++ следует использовать нотацию doxygen [2].

Методические указания для самостоятельной работы

При подготовке к лабораторной работе необходимо:

- Ознакомится с функциями для работы с процессами.
- Ознакомится с функциями для работы с потоками.
- Ознакомиться с функциями для работы с файлами.
- Ознакомиться с функциями для работы с общими ресурсами.
- Ознакомиться с функциями межпроцессного взаимодействия.
- Ознакомиться с функциями стандартной библиотеки.

- Ознакомиться с форматом файлов логов web-сервера Apache.

Алгоритм

Данную задачу следует решить несколькими способами.

Первый подход, который реализует поставленную задачу, описан ниже.

Основной поток

1. Разбор ключей командной строки.

2. Подготовка структур для управления совместным доступом к результирующим файлам.

3. Создание указанного количества результирующих файлов-логов.

4. Запуск указанного количества потоков чтения файлов-логов и записи считанных строк в результирующие файлы после слияния.

Поток чтения-записи

Изначально, потоку передается список (стек) имен файлов, которые ему необходимо проанализировать. После окончания анализа первого файла из списка, поток переходит к обработке следующего, и так до тех пор, пока не будут обработаны все файлы из списка (стека). Такой подход планирования нагрузки не является оптимальным, однако он существенно упрощает задачу лабораторной работы.

1. Открытие файла-лога для чтения. Чтение осуществляется в НЕ ЭКСКЛЮЗИВНОМ режиме. Проверка корректности открытия.

2. Открытие всех результирующих файлов-логов для записи. Запись осуществляется в НЕ ЭКСКЛЮЗИВНОМ режиме. Проверка корректности открытия.

3. Построчное считывание разбор и запись строк-логов.

3.1. Разбор строки.

3.2. Определение результирующего файла-лога, в который необходимо занести считанную строку.

3.3. Занесение строки в результирующий файл. Необходимо учитывать, что одновременно данный файл может быть уже использован другим потоком для записи – организовать совместный доступ.

4. Закрытие файла-лога, открытого для чтения.

5. Закрытие всех результирующих файлов-логов, открытых для записи.

6. Завершение потока.

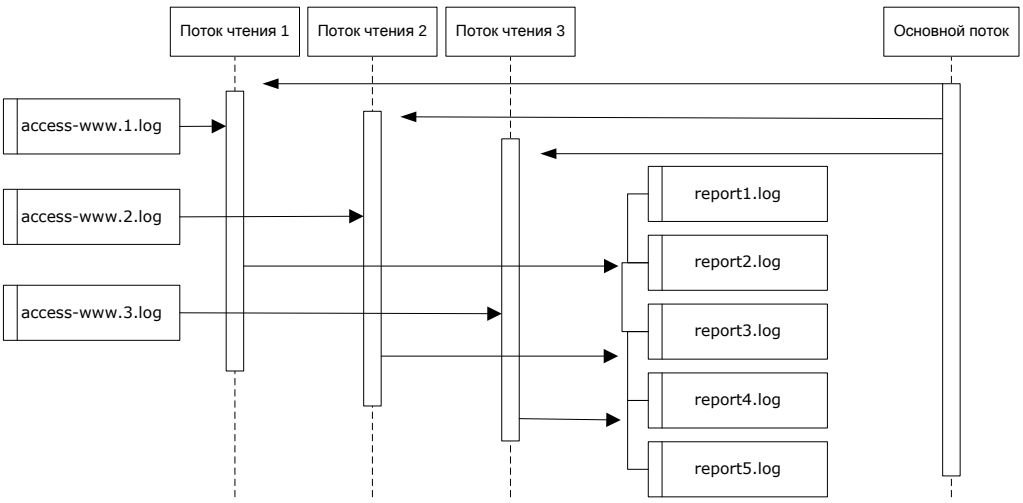

Рис. 1. Диаграмма последовательностей

Второй подход аналогичен первому, отличие состоит лишь в использовании процессов, вместо потоков. Причем обмен информацией между процессами следует организовать, используя средства IPC.

Тестирование

Тестирование приложения осуществляется в несколько этапов:

- Разработка Unit Test в рамках проекта. Данный подход позволит проверить корректность реализации алгоритма.

Теоретические сведения

Необходимая информация по Windows платформе доступна в [1].

Литература

1. Microsoft Developer Network. URL: http://www.msdn.com

2. Формирование документации к исходному коду с помощью средства doxygen. URL: www.nrjetix.com/r-and-d/lectures

3. Описание формата лога web-сервера Apache доступно по адресу: http://www.oglib.ru/apman/logs.html

Приложение

Пример лога web-сервера Apache приведен ниже:

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/bg_foot_line.gif HTTP/1.1" 200 2281 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/bg_footer.jpg HTTP/1.1" 200 3430 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/icons/icon_bug.gif HTTP/1.1" 200 1149 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/logo_footer.gif HTTP/1.1" 200 4354 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /favicon.ico HTTP/1.1" 200 990 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

109.184.36.140 - - [15/Feb/2010:17:40:20 +0200] "GET /index/player_code/?v=688 HTTP/1.1" 302 26 "http://www.clubteatro.ru/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.4 (build 02652) Firefox/3.5.7"