

УДК 004.056.2

**УПРАВЛЕНИЕ ТРАНЗАКЦИЯМИ НАД ФАЙЛОВЫМИ СИСТЕМАМИ
В JAVA-ПРИЛОЖЕНИЯХ****В.А. МАКАРЫЧЕВА, М.Ю. МАКАРЫЧЕВ**
(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

Рассматривается пример использования системы XADisk, которая позволяет работать с файловой системой, обеспечивая поддержку транзакций.

Введение. Большинство программ хранят важные данные в файловых системах, базах данных и других хранилищах. Практически любая база данных имеет механизм транзакции. При взаимодействии с файловой системой, механизм транзакции зачастую не применяется, потому как могут возникнуть проблемы, например, с целостностью данных. Во избежание этой и других проблем, связанных с хранением данных на файловых системах, предлагается использовать XADisk.

Основная часть. XADisk – это комплексная транзакционная система с открытым исходным кодом, написанная на языке программирования Java, предназначенная для возможности обеспечить транзакционный доступ к существующим файловым системам для всех видов Java-приложений, работающих на серверах Web-приложений, приложений JavaEE и т.д. [1].

При помощи нескольких простых API вы можете организовать XADisk поверх любой виртуальной машины Java (JVM) без необходимости установки на операционную систему (ОС). После запуска приложения будут вызывать API XADisk для выполнения различных операций ввода-вывода с файлами в рамках транзакций.

Сама система XADisk представляет собой промежуточную ступень между Java-приложениями и файловыми системами. Продукт XADisk не является реализацией файловой системы; он может работать с самыми разнообразными файловыми системами (например, NTFS, FAT, ext3), обеспечивая возможность транзакционного доступа приложений к этим файловым системам. Приложения вызывают API XADisk для выполнения различных операций ввода-вывода и могут применять команды commit (фиксация) или rollback (откат) для всех этих операций в рамках единой транзакции [2].

Для примера применения XADisk можно рассмотреть некое Java-приложение. Этому приложению нужно создать новый файл с именем file1, записать определенные данные в этот файл из другого файла с именем file2, а затем удалить файл file2. Эти три операции могут быть выполнены в рамках одной транзакции с использованием XADisk, при этом соблюдаются такие свойства, как:

- 1) целостность;
- 2) согласованность;
- 3) изолированность;
- 4) долговечность.

XADisk будет предоставлять эти гарантии даже в том случае, если во время процесса произойдет аварийное завершение работы приложения.

XADisk обладает следующими операциями:

1. с файлами:
 - 1.1. create (создание);
 - 1.2. delete (удаление);
 - 1.3. move (перемещение);
 - 1.4. copy (копирование);
 - 1.5. truncate (усечение);
 - 1.6. read (чтение);
 - 1.7. write (запись);
 - 1.8. exists (проверка существования);
 - 1.9. getLength (возврат длины);
2. с каталогами:
 - 2.1. create (создание);
 - 2.2. delete (удаление);
 - 2.3. move (перемещение);
 - 2.4. list children (список потомков);
 - 2.5. exists (проверка существования).

При эксплуатации XADisk не требуется, чтобы приложения и файловая система находились на одной JVM. Поэтому приложение может удаленно вызвать API XADisk, как и сам XADisk, расположенный на той же JVM.

Перед тем, как вызвать операции ввода/вывода через XADisk, необходимо загрузить экземпляр XADisk на JVM. На одной виртуальной машине может располагаться не один экземпляр XADisk, но при этом, каждый экземпляр XADisk должен иметь собственное изолированное состояние.

После того, как экземпляр XADisk загружен и готов к работе, можно выполнять транзакционных операций с файлами и каталогами. Так как данные операции будут осуществляться внутри транзакции, то необходимо создать сеанс для транзакции. Только после установки сеанса можно переходить к выполнению операций ввода/вывода.

После завершения всех операций можно выполнить фиксацию (или откат) транзакции с помощью простого вызова сеансового объекта. Для новой транзакции необходимо создать новый сеансовый объект. Работу экземпляра XADisk можно завершить из любого места в JVM.

Примера использования XADisk представлен в листинге 1.

Листинг 1 – код работы XADisk

```

1: String XADiskSystemDirectory = "/home/systems/XADiskSystem1";
2: StandaloneFileSystemConfiguration configuration =
3: new StandaloneFileSystemConfiguration(XADiskSystemDirectory, "instance-1");
4: XAFileSystem xaf = XAFileSystemProxy.bootNativeXAFileSystem(configuration);
5: xaf.waitForBootup(10000L);
6: XAFileSystem xaf = XAFileSystemProxy.getNativeXAFileSystemReference("instance-1");
7: XAFileSystem xaf = XAFileSystemProxy.getRemoteXAFileSystemReference("10.30.9.200", 5151);
8: Session session = xaf.createSessionForLocalTransaction();
9: File f = new File("/testAPIs/test.txt");
10: if(session.fileExists(f)) {
11:     XAFileInputStream xis = session.createXAFileInputStream(f);
12:     for (int i = 0; i < 100; i++) {
13:         byte a = (byte) xis.read();
14:         if( a== -1) {
15:             break;
16:         }
17:         System.out.print(a);
18:     }
19:     xis.close();
20:     session.moveFile(f, new File("/testAPIs/test.txt____" + System.currentTimeMillis()));
21: }
22: else {
23:     session.createFile(f, false);
24:     XAFileOutputStream xafos = session.createXAFileOutputStream(f, false);
25:     byte[] buffer = new byte[100];
26:     for (int i = 0; i < 100; i++) {
27:         buffer[i] = i*i;
28:     }
29:     xafos.write(buffer);
30:     xafos.close();
31: }
32: //фиксация транзакции
33: session.commit();
34: //или откат транзакции
35: session.rollback();

```

Заключение. Реализация доступа к файловым системам с использованием механизма транзакций может предложить множество преимуществ для приложений, которые хранят свои данные в файловых системах, а именно целостность: согласованность; изолированность; долговечность.

ЛИТЕРАТУРА

1. stackoverflow [Электронный ресурс]. – XADisk. – Режим доступа: <https://stackoverflow.com/tags/xadisk/info>. – Дата доступа: 20.09.18.
2. IBM [Электронный ресурс] // Транзакционный доступ к файловым системам с использованием XADisk. – Режим доступа: <https://www.ibm.com/developerworks/ru/library/os-xadisk/>. – Дата доступа: 20.09.18.