

priority 键的报头。

对于每个优先级为 1 的事件写入到三个 Channel，c1、c2 和 c3，c3 被标记为是可选的。所以，如果写入到 c1 或 c2 失败，Source 会从 Channel 处理器的 processEvent 或 processEventBatch 方法得到一个异常，且该 Source 必须重试。但是因为 c3 标记为可选的，如果写入到 c3 失败，Source 不会得到异常且意识不到失败，因为这个失败被 Channel 处理器所忽略。

155

类似地，任何优先级为 2 的事件写入到 c2 和可选的 c4。可以清楚地看出，Channel 可以出现在多个映射中，例如这个例子中的 c2。优先级为 1 或 2 的 Channel 写入到 c2——这就是图 6-1 展示的例子是如何实现的。

缺少优先级报头的事件，或值不是 1 或 2 的事件，写入到默认的 Channel，在这个例子中是 c5。如果一个事件没有找到所需的 Channel，那么事件将写入到该事件可选的 Channel 和默认的 Channel 中。在这个例子中，优先级为 3 的事件将写入到 Channel c4 和 c5。如果写入到 c4 失败，那么该失败被忽略，但是如果写入到 c5 失败，Source 会得到一个异常，且该事件必须被重新写入。

自定义 Channel 选择器 *

可以编写和部署自定义 Channel 选择器，允许你使用指定部署逻辑来控制事件流。为了实现一个自定义 Channel 选择器，该选择器需要实现 ChannelSelector 接口或继承 AbstractChannelSelector 类。例 6-3 展示了 AbstractChannelSelector 类。

对于每个事件，Channel 处理器调用 Channel 选择器的 getRequiredChannels 和 getOptionalChannels 方法，返回需要的和可选的将要写入事件的 Channel 列表。如果写入到任何所需的 Channel 的操作失败，Channel 处理器将会抛出 ChannelException 异常，并使 Source 重试。任何写入到可选的 Channel 的操作失败都会被忽略。

例6-3 自定义选择器继承的AbstractChannelSelector类

```
package org.apache.flume.channel;

public abstract class AbstractChannelSelector implements ChannelSelector {

    private List<Channel> channels;
    private String name;

    @Override
    public List<Channel> getAllChannels() {
        return channels;
    }
}
```