自：https://blog.csdn.net/heyutao007/article/details/72773077

**log4j2异步Logger**

2017年05月26日 19:52:20 [和大黄](https://me.csdn.net/HEYUTAO007) 阅读数：4091更多

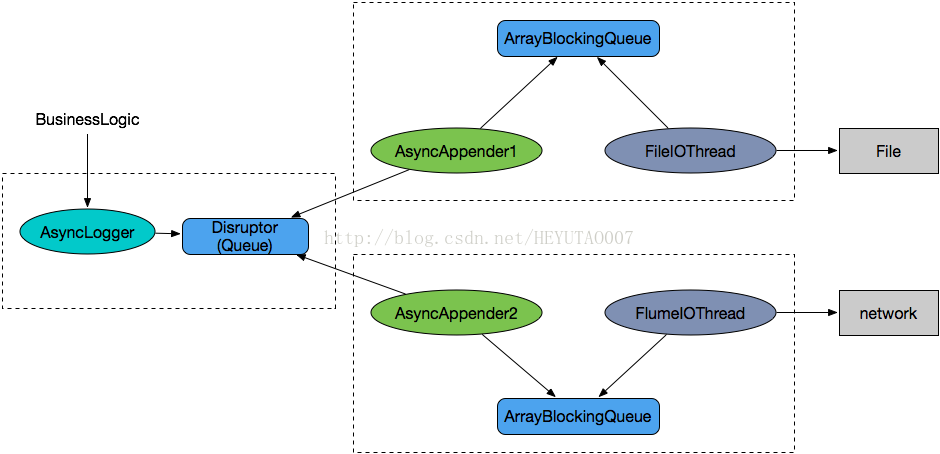
个人分类： [Java](https://blog.csdn.net/HEYUTAO007/article/category/671411)[Web](https://blog.csdn.net/HEYUTAO007/article/category/680606)

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/HEYUTAO007/article/details/72773077

## 1 异步Logger的意义

之前的日志框架基本都实现了AsyncAppender，被证明对性能的提升作用非常明显。

在log4j2日志框架中，增加了对Logger的异步实现。那么这一步的解耦，意义何在呢？

  
如图，按我目前的理解：异步Logger是让业务逻辑把日志信息放入Disruptor队列后可以直接返回（无需等待“挂载的各个Appender”都取走数据）  
  
优点：更高吞吐、调用log方法更低的延迟。  
缺点：异常处理麻烦、 可变日志消息问题、更大的CPU开销、需要等待“最慢的Appender”消费完成。  
异步Logger会使用Disruptor做高吞吐队列，异步Appender会使用ArrayBlockingQueue做队列。  
异步Logger与异步Appender都可以设置“队列满了之后的策略”，我个人建议都设置为“按日志级别部分丢弃”。

## 2 启用

1，全局启用异步Logger方案一  
JVM启动参数（boot.ini）加上“-DLog4jContextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector”  
2，全局启用异步Logger方案二  
classpath中添加文件“log4j2.component.properties”，文件增加以下内容：  
“Log4jContextSelector=org.apache.logging.log4j.core.async.AsyncLoggerContextSelector”  
3，手工指定部分Logger采用异步方式  
log4j2.xml配置文件中使用AsyncRoot/AsyncLogger替代Root/Logger  
（上述3种方式任选其一即可，不要同时采用）