|  |
| --- |
| Neusoft |
| logback |
| Logback自学笔记 |

|  |
| --- |
| 成阳阳  2014/6/10 |
|  |

## 前言

首先介绍下logback的前世今生。

slf4j由log4j作者Ceki开发，逐步取代apahce commons logging。

logback由log4j作者Ceki开发，逐步取代log4j。

slf4j等于commons-logging，是各种日志实现的通用入口，会根据classpath中存在下面哪一个Jar来决定具体的日志实现库。

logback相比较log4j的优势

slf4j支持参数化的logger.error("帐号ID：{}不存在", userId);

告别了if(logger.isDebugEnable()) 时代。

另外logback的整体性能比log4j也较佳，hibernate等项目已经采用了slf4j。

slf4j和logback的使用

1.如果日志的参数超过3个，需要写成

Java代码  [收藏代码](javascript:void())

Object[] params = {newVal, below, above};

logger.debug("Value {} was inserted between {} and {}.", params);

2.因为内部已优化，作者认为slf4j的logger不需要定义为static。

3.可设置缓存后批量写日志文件(但服务器如果重启，可能会丢失未写到磁盘的记录)

4.MDC，用Filter，将当前用户名等业务信息放入MDC中，在日志format定义中即可使用该变量。

5.JMS Appender用于告警, DB Appender用于业务日志等可以使用插件，如生成Log代码的Eclipse插件Log4E。

6.tomcat和glassfish中，设定日志路径为../logs/xxxx.log 都能将日志放入应用服务器本身的logs目录。

logback取代 log4j的理由：

Logback和log4j是非常相似的，如果你对log4j很熟悉，那对logback很快就会得心应手。下面列了logback相对于log4j的一些优点：

1、更快的实现 Logback的内核重写了，在一些关键执行路径上性能提升10倍以上。而且logback不仅性能提升了，初始化内存加载也更小了。

2、非常充分的测试 Logback经过了几年，数不清小时的测试。Logback的测试完全不同级别的。在作者的观点，这是简单重要的原因选择logback而不是log4j。

3、Logback-classic非常自然实现了SLF4j Logback-classic实现了 SLF4j。在使用SLF4j中，你都感觉不到logback-classic。而且因为logback-classic非常自然地实现了SLF4J， 所 以切换到log4j或者其他，非常容易，只需要提供成另一个jar包就OK，根本不需要去动那些通过SLF4JAPI实现的代码。

4、非常充分的文档 官方网站有两百多页的文档。

5、自动重新加载配置文件 当配置文件修改了，Logback-classic能自动重新加载配置文件。扫描过程快且安全，它并不需要另外创建一个扫描线程。这个技术充分保证了应用程序能跑得很欢在JEE环境里面。

6、Lilith Lilith是log事件的观察者，和log4j的chainsaw类似。而lilith还能处理大数量的log数据 。

7、谨慎的模式和非常友好的恢复 在谨慎模式下，多个FileAppender实例跑在多个JVM下，能 够安全地写道同一个日志文件。RollingFileAppender会有些限制。Logback的FileAppender和它的子类包括 RollingFileAppender能够非常友好地从I/O异常中恢复。

8、配置文件可以处理不同的情况 开发人员经常需要判断不同的Logback配置文件在不同的环境下（开发，测试，生产）。而这些配置文件仅仅只有一些很小的不同，可以通过,和来实现，这样一个配置文件就可以适应多个环境。

9、Filters（过滤器） 有些时候，需要诊断一个问题，需要打出日志。在log4j，只有降低日志级别，不过这样会打出大量的日志，会影响应用性能。在Logback，你可以继续 保持那个日志级别而除掉某种特殊情况，如alice这个用户登录，她的日志将打在DEBUG级别而其他用户可以继续打在WARN级别。要实现这个功能只需 加4行XML配置。可以参考MDCFIlter 。

10、SiftingAppender（一个非常多功能的Appender） 它可以用来分割日志文件根据任何一个给定的运行参数。如，SiftingAppender能够区别日志事件跟进用户的Session，然后每个用户会有一个日志文件。

11、自动压缩已经打出来的log RollingFileAppender在产生新文件的时候，会自动压缩已经打出来的日志文件。压缩是个异步过程，所以甚至对于大的日志文件，在压缩过程中应用不会受任何影响。

12、堆栈树带有包版本 Logback在打出堆栈树日志时，会带上包的数据。

13、自动去除旧的日志文件 通过设置TimeBasedRollingPolicy或者SizeAndTimeBasedFNATP的maxHistory属性，你可以控制已经产生日志文件的最大数量。如果设置maxHistory 12，那那些log文件超过12个月的都会被自动移除。

总之，logback比log4j太优秀了，让我们的应用全部建立logback上吧 ！